

JEFF KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

| REPORT DOCUMENTATION PAGE | | | |
|--|--|--|------------------------------|
| 1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED | | 1b. RESTRICTIVE MARKINGS | |
| 2a. SECURITY CLASSIFICATION AUTHORITY | | 3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited. | |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | | | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) | | 5. MONITORING ORGANIZATION REPORT NUMBER(S) | |
| 6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School | 6b. OFFICE SYMBOL (If applicable) 39 | 7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School | |
| 6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000 | | 7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000 | |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION | 8b. OFFICE SYMBOL (If applicable) | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER | |
| 8c. ADDRESS (City, State, and ZIP Code) | | 10. SOURCE OF FUNDING NUMBERS | |
| | | Program Element No. | Project No. |
| | | Task No. | Work Unit Accession Number |
| 11. TITLE (Include Security Classification) THERMOELECTRIC COOLER DESIGN | | | |
| 12. PERSONAL AUTHOR(S) Clifton, William E. | | | |
| 13a. TYPE OF REPORT Master's Thesis | 13b. TIME COVERED From To | 14. DATE OF REPORT (year, month, day) December 1992 | 15. PAGE COUNT 79 |
| 16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. | | | |
| 17. COSATI CODES | | 18. SUBJECT TERMS (continue on reverse if necessary and identify by block number) | |
| FIELD | GROUP | SUBGROUP | |
| | | Thermoelectric Cooler, Peltier Cooler, Seebeck Effect, Thomson Effect, Thermocouple, Coefficient of Performance, Windows | |
| | | | |
| 19. ABSTRACT (continue on reverse if necessary and identify by block number) | | | |
| Thermoelectric Cooler Design is a Microsoft Windows program to be used as an aid in the design of thermoelectric cooler devices. This program was written to be used to quickly model and compare alternative designs. A couple's optimum coefficient of performance and maximum heat pumping can be quickly determined. Other major features of the program include the ability to change material properties and dimensions of couples, analyze cascaded couples, and graph performance parameters. A brief description of thermoelectric cooler theory, modelling assumptions and complete source code listing is included. | | | |
| | | | |
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS REPORT <input type="checkbox"/> DTIC USERS | | 21. ABSTRACT SECURITY CLASSIFICATION Unclassified | |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL Allan D. Kraus | | 22b. TELEPHONE (Include Area code) (408) 656-2730 | 22c. OFFICE SYMBOL ECE-KS |

Approved for public release; distribution is unlimited.

Thermoelectric Cooler Design

by

William Earle Clifton
Lieutenant Commander, United States Navy
B.A., University of California at Los Angeles, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING
(SPACE SYSTEMS ENGINEERING)

from the

NAVAL POSTGRADUATE SCHOOL
December 1992

TABLE OF CONTENTS

| | |
|--|----|
| I. INTRODUCTION | 1 |
| A. Features Of Thermoelectric Coolers | 1 |
| B. Spaceborne Applications | 1 |
| II. THERMOELECTRIC COOLER THEORY | 3 |
| A. Thermoelectric Couple Configuration | 3 |
| B. Thermoelectric Effects | 4 |
| 1. Seebeck Effect | 4 |
| 2. Peltier Effect | 5 |
| 3. Thomson Effect | 5 |
| C. Irreversible Effects | 6 |
| 1. Joule Effect | 6 |
| 2. Fourier Effect | 7 |
| D. Net Heat Absorbed At Cold Junction Of Couple | 8 |
| III. THERMOELECTRIC COOLER MODEL | 15 |
| A. Introduction | 15 |
| B. Heat Pumping | 15 |
| 1. Current Yielding Maximum Heat Pumped | 15 |
| 2. Maximum Heat Pumped | 16 |
| C. Maximum Temperature Differential | 16 |
| D. Optimum Ratio Of Element Cross Sectional Areas | 16 |
| E. Figure Of Merit Of Element Materials | 19 |
| F. Power Required To Operate The Couple | 20 |
| G. Coefficient Of Performance | 20 |
| 1. Coefficient Of Performance At Maximum Heat Pumping | 20 |
| 2. Current Yielding Optimum Coefficient Of Performance | 22 |
| 3. Optimum Coefficient Of Performance | 25 |
| H. Cascaded Coolers | 25 |
| 1. Overall Heat Pumped By Cascaded Coolers | 27 |
| 2. Overall Coefficient Of Performance | 27 |
| I. Summary Of Simplifying Assumptions | 27 |

1780
C5137
C.4

| | |
|--|----|
| 1. Thomson Voltage | 27 |
| 2. Temperature Losses | 28 |
| 3. Parallel Thermal Paths | 28 |
| 4. Junction Resistance | 28 |
| 5. Figure Of Merit | 28 |
| 6. Resistivity And Conductivity | 28 |
| 7. Couple Power Supply | 28 |
| IV. THERMOELECTRIC COOLER DESIGN PROGRAM | 29 |
| A. Introduction | 29 |
| 1. Features | 29 |
| 2. System Requirements | 30 |
| 3. Disk Contents | 30 |
| 4. Installation | 30 |
| B. Using The Program | 31 |
| 1. Starting And Exiting The Program | 31 |
| 2. Setting Material Properties | 32 |
| 3. Optimum Coefficient Of Performance | 33 |
| 4. Maximum Heat Pumping | 34 |
| 5. Cascading Stages | 35 |
| 6. Graphing The Coefficient Of Performance And Heat Pumped | 36 |
| 7. Temperature Conversion Calculator | 39 |
| 8. Diagrams And Glossary | 40 |
| V. CONCLUSIONS | 41 |
| APPENDIX | 42 |
| LIST OF REFERENCES | 70 |
| BIBLIOGRAPHY | 71 |
| INITIAL DISTRIBUTION LIST | 72 |

ABSTRACT

Thermoelectric Cooler Design is a Microsoft Windows program to be used as an aid in the design of thermoelectric cooler devices. This program was written to be used to quickly model and compare alternative designs. A couple's optimum coefficient of performance and maximum heat pumping can be quickly determined. Other major features of the program include the ability to change material properties and dimensions of couples, analyze cascaded couples, and graph performance parameters. A brief description of thermoelectric cooler theory, modelling assumptions and complete source code listing is included.

I. INTRODUCTION

A. FEATURES OF THERMOELECTRIC COOLERS

The unique features of thermoelectric coolers include:

1. Compact.
2. No moving parts.
3. The coefficient of performance is independent of capacity.
4. The cooling rate is easily controlled.
5. Reversible operation is possible allowing heating as well as cooling.
6. Vibration-free operation.
7. Precise temperature control.

B. SPACEBORNE APPLICATIONS

Thermoelectric coolers offer an alternative solution to cooling and heating requirements of spacecraft. Spaceborne applications for thermoelectric coolers include:

1. Equipment cooling -- removal of waste heat in order to maintain components at their optimal operating temperature. Thermoelectric coolers are especially useful for the control of small hot spots.

2. Life support temperature control -- thermoelectric coolers are very reliable making them ideally suited for long duration missions such as the space station or manned Mars mission.
3. Cooling of infrared detectors -- thermoelectric coolers eliminate the need for coolant and associated support equipment. Of even more importance is the elimination of consumable coolants which also increases the life of the spacecraft.
4. Microgravity applications -- thermoelectric coolers offer precise, vibration-free temperature control for free-flying orbiting microgravity research platforms.

Reliability, the lack of moving parts or consumables, and vibration-free operation are especially important to spaceborne cooling systems.

II. THERMOELECTRIC COOLER THEORY

A. THERMOELECTRIC COUPLE CONFIGURATION

In its most basic form, a thermoelectric couple consists of two dissimilar elements with contact at two junctions. Figure 1 shows a simple thermoelectric circuit consisting of two elements, physically joined on one side while electrically joined at the other.

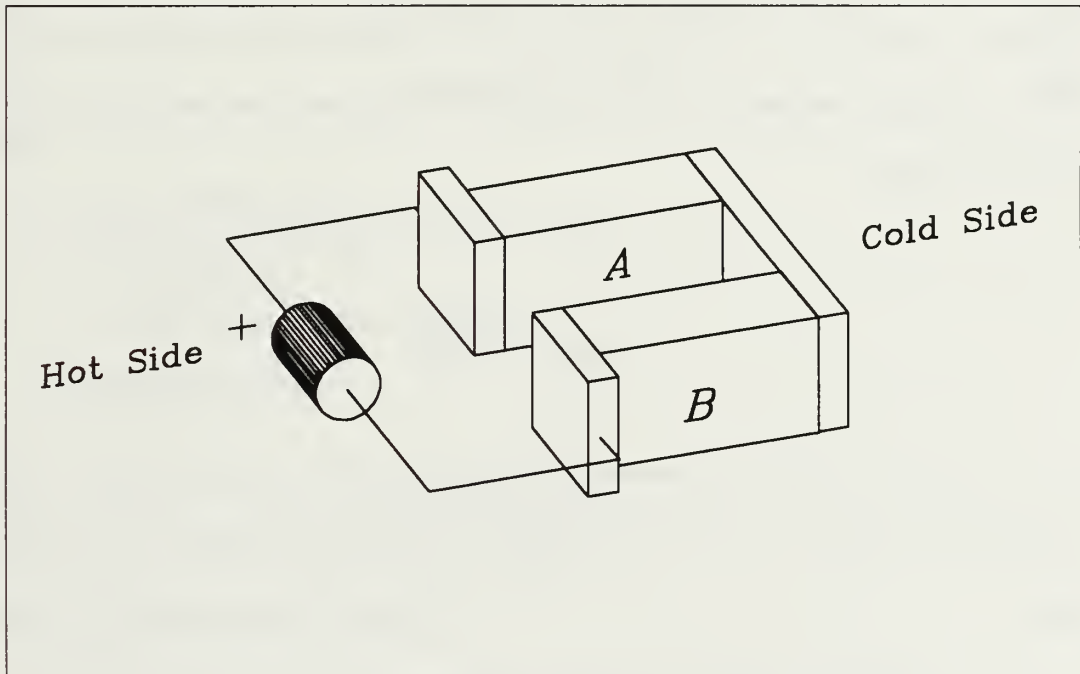


Figure 1 Thermoelectric Cooler

B. THERMOELECTRIC EFFECTS

The background for this section was gathered from Kraus [Ref. 12:pp. 294-303] and Ioffe [Ref. 2:pp. 96-128]. Three thermoelectric effects are known; the Seebeck, Peltier, and Thomson effects.

1. Seebeck Effect

Consider the junction of two materials, A and B . If a temperature differential exists between the two junctions, one junction may be referred to as the hot side and the other the cold side. An open circuit voltage can be measured between the two junctions. This net conversion of thermal energy into electrical energy under zero current conditions is known as the Seebeck effect. The voltage observed, referred to as the Seebeck voltage, is proportional to the temperature differential between the hot and cold sides of the couple:

$$dE_S \propto dT$$

$$dE_S = \pm \alpha dT$$

$$E_S = \pm \int_{T_1}^{T_2} \alpha dT \quad (\text{V}) \quad (1)$$

where

E_S = Seebeck voltage (V)

α = Seebeck coefficient of material ($\text{V}/^\circ\text{C}$)

In the case of the couple composed of materials A and B ,

$$\alpha = |\alpha_A| + |\alpha_B|$$

where

α_A = Seebeck coefficient of material A

α_B = Seebeck coefficient of material B

2. Peltier Effect

The Peltier effect is localized at the junctions between materials A and B . When a current flows through the junction, heat is either generated or absorbed depending on the direction of the current flow. The amount of heat is proportional to the current and is known as the Peltier heat:

$$dQ_p \propto I dt$$

$$dQ_p = \pm \pi I dt$$

$$q_p = \pm \pi I \quad (\text{W}) \quad (2)$$

where

Q_p = Peltier heat (J)

q_p = Peltier heat flow (W)

I = Current (A)

π = Peltier coefficient of material (V)

3. Thomson Effect

This effect concerns the absorption or generation of heat in a conductor carrying a current in the presence of a temperature gradient. The amount of heat is proportional to the current and temperature gradient:

$$dQ_t \propto I dT dt$$

$$dQ_t = \pm \sigma I dT dt$$

$$dq_t = \pm \sigma I dT$$

$$q_t = \pm \sigma I \int_{T_1}^{T_2} dT \quad (\text{W}) \quad (3)$$

where

Q_t = Thomson heat (J)

q_t = Thomson heat (W)

I = Current (A)

σ = Thomson coefficient of material ($\text{V}/^\circ\text{C}$)

Because the product of σ and ΔT is a voltage, the Thomson voltage is therefore:

$$E_t = \pm \int_{T_1}^{T_2} \sigma dT \quad (\text{V}) \quad (4)$$

C. IRREVERSIBLE EFFECTS

1. Joule Effect

The evolution of heat when a current passes through a material may be referred to as the Joule heat flow:

$$q_j = I^2 R \quad (\text{W}) \quad (5)$$

In the case of a couple, the elements A and B are connected in electrical series regardless of the direction of the current flow:

$$R = \frac{\rho_A L_A}{A_A} + \frac{\rho_B L_B}{A_B} \quad (\Omega) \quad (6)$$

where

ρ_A, ρ_B = electrical resistivity of materials A and B ($\Omega \text{ cm}$)

L_A, L_B = Length of materials A and B (cm)

A_A, A_B = Cross sectional area of materials A and B (cm^2)

2. Fourier Effect

The conduction of heat through a material is given by:

$$q_f = \left(\frac{K A}{L} \right) \Delta T \quad (\text{W}) \quad (7)$$

and, in the case of a couple with the elements in thermal parallel,

$$K = \frac{k_A A_A}{L_A} + \frac{k_B A_B}{L_B} \quad (\text{W}/^\circ\text{C}) \quad (8)$$

where

k_A, k_B = Thermal conductivity of materials A and B ($\text{W}/\text{cm } ^\circ\text{C}$)

L_A, L_B = Length of materials A and B (cm)

A_A, A_B = Cross sectional area of materials A and B (cm^2)

D. NET HEAT ABSORBED AT COLD JUNCTION OF COUPLE

In this analysis, the Thomson heat will be neglected which greatly simplifies couple design procedure without any significant error Kraus [Ref. 1:p. 300] and Cadoff [Ref. 3:p. 8]. To determine the net heat pumped by the couple, the temperature distribution along the elements of the couple must be determined. Consider a single thermoelectric element as indicated in Figure 2, where A_x is the cross sectional area of the element normal to the flow of current and heat.

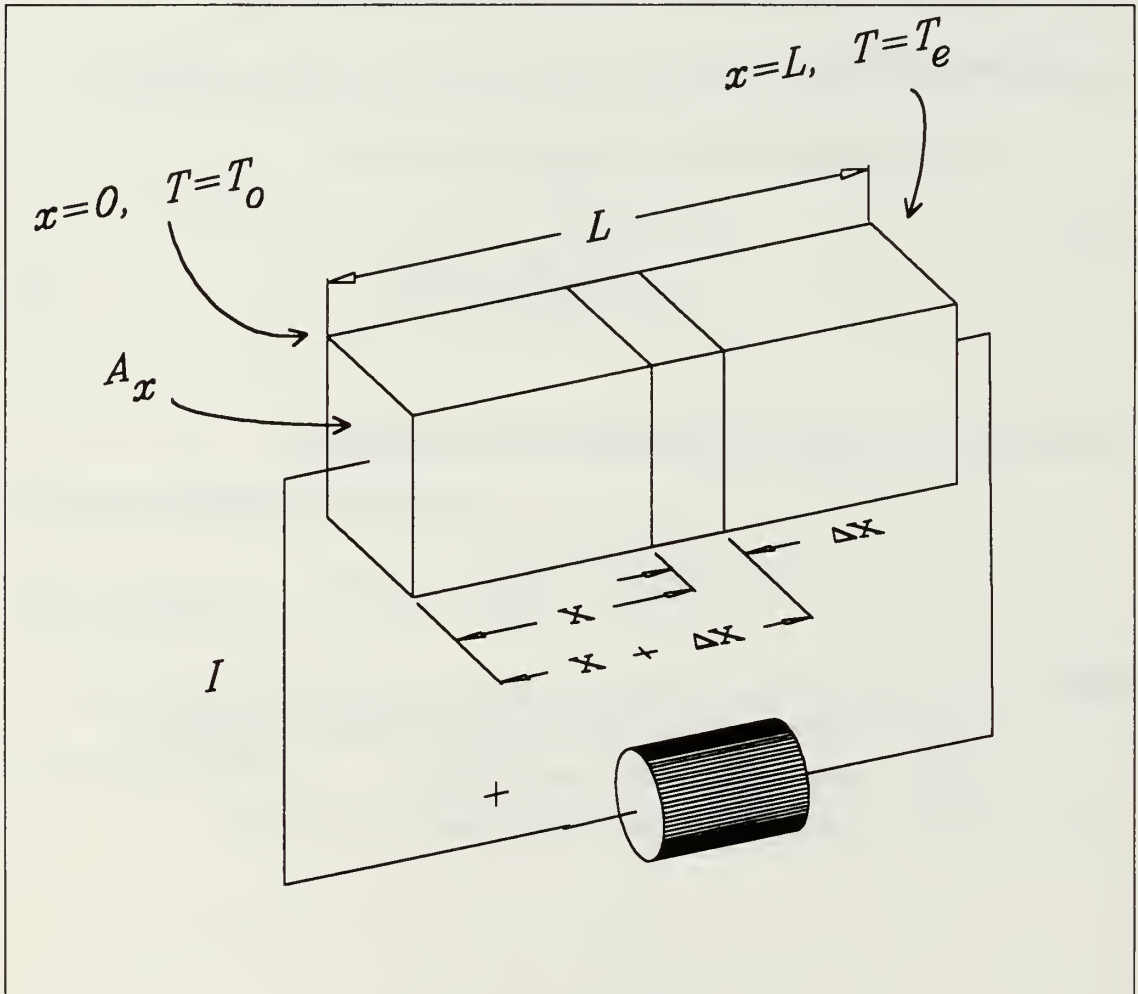


Figure 2 Thermoelectric Element

Here, the boundary conditions are:

$$\begin{aligned}T &= T_o \text{ at } x = 0 \\T &= T_e \text{ at } x = L\end{aligned}$$

Assuming steady state temperature conditions, there are three heat quantities which pertain to an energy balance over the incremental length, Δx :

1. The Fourier heat entering at point x :

$$q_x = -k A_x \left[\frac{dT}{dx} \right]_x$$

2. The Joulean heat generated in the increment Δx :

$$q_j = \frac{I^2 \rho \Delta x}{A_x}$$

3. The Fourier heat leaving at point $x + \Delta x$:

$$q_{x+\Delta x} = -k A_x \left[\frac{dT}{dx} \right]_{x+\Delta x}$$

The energy balance in the steady state requires that the heat leaving must be equal to the sum of the heat entering and the heat generated in the increment Δx :

$$q_x + q_j = q_{x+\Delta x}$$

Substituting for q_x , $q_{x+\Delta x}$ and q_j gives

$$-kA_x \left[\frac{dT}{dx} \right]_x + \frac{I^2 \rho \Delta x}{A_x} = -kA_x \left[\frac{dT}{dx} \right]_{x+\Delta x}$$

or with some algebra,

$$kA_x \left(\left[\frac{dT}{dx} \right]_{x+\Delta x} - \left[\frac{dT}{dx} \right]_x \right) + \frac{I^2 \rho \Delta x}{A_x} = 0$$

Dividing through by Δx yields

$$kA_x \left(\frac{\left[\frac{dT}{dx} \right]_{x+\Delta x} - \left[\frac{dT}{dx} \right]_x}{\Delta x} \right) + \frac{I^2 \rho}{A_x} = 0$$

And taking the limit as $\Delta x \rightarrow 0$ gives

$$\lim_{\Delta x \rightarrow 0} \left[kA_x \left(\frac{\left[\frac{dT}{dx} \right]_{x+\Delta x} - \left[\frac{dT}{dx} \right]_x}{\Delta x} \right) + \frac{I^2 \rho}{A_x} \right] = 0$$

which can be adjusted to

$$kA_x \frac{d^2 T}{dx^2} + \frac{I^2 \rho}{A_x} = 0 \quad (9)$$

The general solution to this simple differential equation is obtained through the use of a double integration:

$$\frac{d^2 T}{dx^2} + \frac{I^2 \rho x}{kA_x^2} + C_1 = 0$$

and

$$T + \frac{I^2 \rho x^2}{2kA_x^2} + C_1 x + C_2 = 0$$

Applying the boundary conditions for the hot side where $x = 0$ and $T = T_o$:

$$C_2 = -T_o$$

and

$$T - T_o + \frac{I^2 \rho x^2}{2kA_x^2} + C_1 x = 0$$

At the cold side of the element where $x = L$ and $T = T_e$:

$$T_e - T_o + \frac{I^2 \rho L^2}{2kA_x^2} + C_1 L = 0$$

The solution for C_1 , the first constant of integration, is seen to be

$$C_1 = \frac{T_o - T_e}{L} - \frac{I^2 \rho L}{2kA_x^2}$$

so that the particular solution of this simple, second order, differential equation which gives the temperature profile as a function of distance along the element:

$$T - T_o + \frac{I^2 \rho x^2}{2kA_x^2} + \left(\frac{T_o - T_e}{L} \right) x - \left(\frac{I^2 \rho L}{2kA_x^2} \right) x = 0$$

or

$$T(x) = -\left(\frac{I^2 \rho}{2kA_x^2}\right)x^2 - \left(\frac{T_o - T_e}{L} - \frac{I^2 \rho L}{2kA_x^2}\right)x + T_o \quad (10)$$

which is seen to be parabolic.

Figure 3 illustrates a temperature distribution for a typical element with the following parameters:

$$z = 3.08 \times 10^{-3} \text{ } (^{\circ}\text{C}^{-1}) \quad (11)$$

$$R = 1.73 \times 10^{-3} \text{ } (\Omega) \quad (12)$$

$$K = 3.64 \times 10^{-2} \text{ } (\text{W}/^{\circ}\text{C}) \quad (13)$$

$$\alpha = 4.4 \times 10^{-4} \text{ } (\text{V}/^{\circ}\text{C}) \quad (14)$$

$$T_c = 308.6 \text{ } \text{K} \quad (15)$$

$$\Delta T = 50 \text{ } ^{\circ}\text{C} \quad (16)$$

Due to the parabolic temperature distribution, the Joulean heat will be transferred to both the cold and hot junctions of the couple.

Differentiating equation (10), setting the derivative equal to zero,

$$\frac{dT}{dx} = -\frac{I^2 \rho X}{kA_x^2} - \frac{T_o - T_e}{L} + \frac{I^2 \rho L}{2kA_x^2} = 0$$

permits a solution for $x = X$ which is the position where the maximum temperature occurs:

$$X = \frac{\frac{I^2 \rho L}{2kA_x^2} - \frac{T_o - T_e}{L}}{\frac{I^2 \rho X}{kA_x^2}}$$

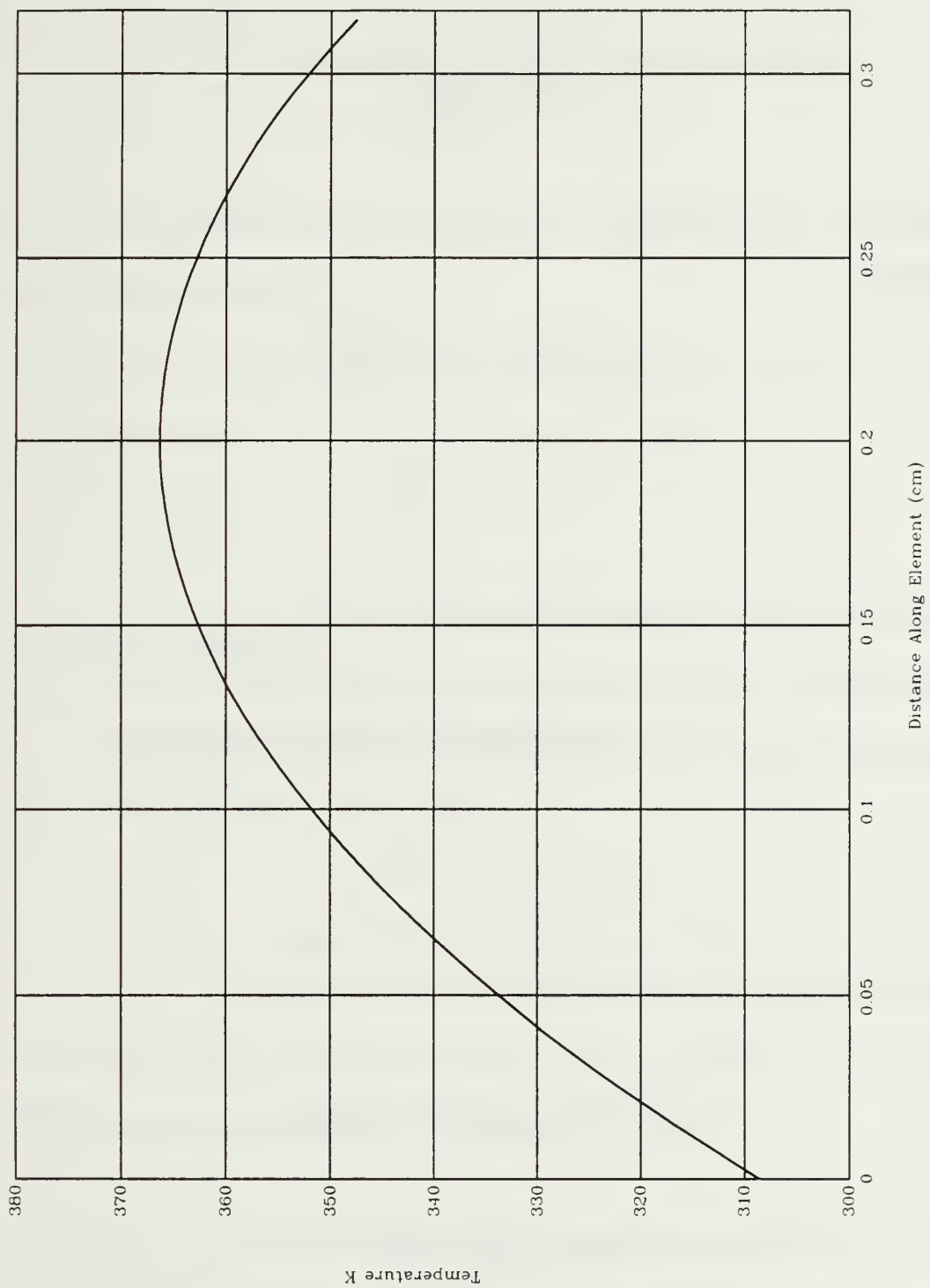


Figure 3 Temperature Profile Along Element

Rearranging and substituting ΔT for $T_o - T_e$:

$$X = \frac{L}{2} - \frac{k \Delta T A_x^2}{I^2 \rho L}$$

Assuming $x = L$ and defining f as the fraction of the Joulean heat transferred to the cold junction:

$$f = \frac{L-X}{L} = \frac{1}{2} + \frac{k \Delta T A_x^2}{I^2 \rho L^2}$$

or

$$f = \frac{1}{2} + \frac{K \Delta T}{I^2 R} \quad (17)$$

where K and R are defined by equations (6) and (8), respectively.

The net heat absorbed at the cold junction of the couple is equal to the summation of the Peltier heat, minus the fraction of Joulean heat transferred to the cold junction:

$$q_{net} = \alpha I T_c - f I^2 R$$

and upon substitution of the expression for f given by equation (17):

$$q_{net} = \alpha I T_c - \frac{1}{2} I^2 R - K \Delta T \quad (W) \quad (18)$$

Therefore the net heat pumped is the Peltier heat developed at the cold junction less the sum of one half the Joulean heat produced in the couples and the heat due to conduction from the hot junction to the cold junction.

III. THERMOELECTRIC COOLER MODEL

A. INTRODUCTION

The equations in this chapter were used to model the thermoelectric cooler couple operating parameters and were given by Kraus [Ref. 1:pp. 303 - 326].

B. HEAT PUMPING

1. Current Yielding Maximum Heat Pumped

The equation for net heat pumped

$$q_{net} = \alpha I T_c - \frac{1}{2} I^2 R - K \Delta T \quad (\text{W}) \quad (18)$$

may be differentiated with respect to current, I . When the derivative is set equal to zero, the current that yields maximum heat pumped is obtained:

$$\frac{dq}{dI} = \alpha T_c - IR = 0$$

Solving for I :

$$I_m = \frac{\alpha T_c}{R} \quad (\text{A}) \quad (19)$$

2. Maximum Heat Pumped

Substituting I_m into the equation for the net heat pumped yields the maximum heat pumped:

$$q_m = \frac{\alpha^2 T_c^2}{2R} - K\Delta T \quad (\text{W}) \quad (20)$$

C. MAXIMUM TEMPERATURE DIFFERENTIAL

Heat pumped as a function of current is shown in Figure 4 using the parameters defined in equations (11) through (16). Setting equation (20) which is the maximum heat pumped equal to zero and solving for ΔT , one obtains the maximum temperature differential between the hot and cold sides of the couple:

$$\Delta T_m = \frac{\alpha^2 T_c^2}{2KR} \quad (\text{K}) \quad (21)$$

D. OPTIMUM RATIO OF ELEMENT CROSS SECTIONAL AREAS

The maximum temperature differential given by equation (21), can be increased by minimizing the product KR from equations (6) and (8):

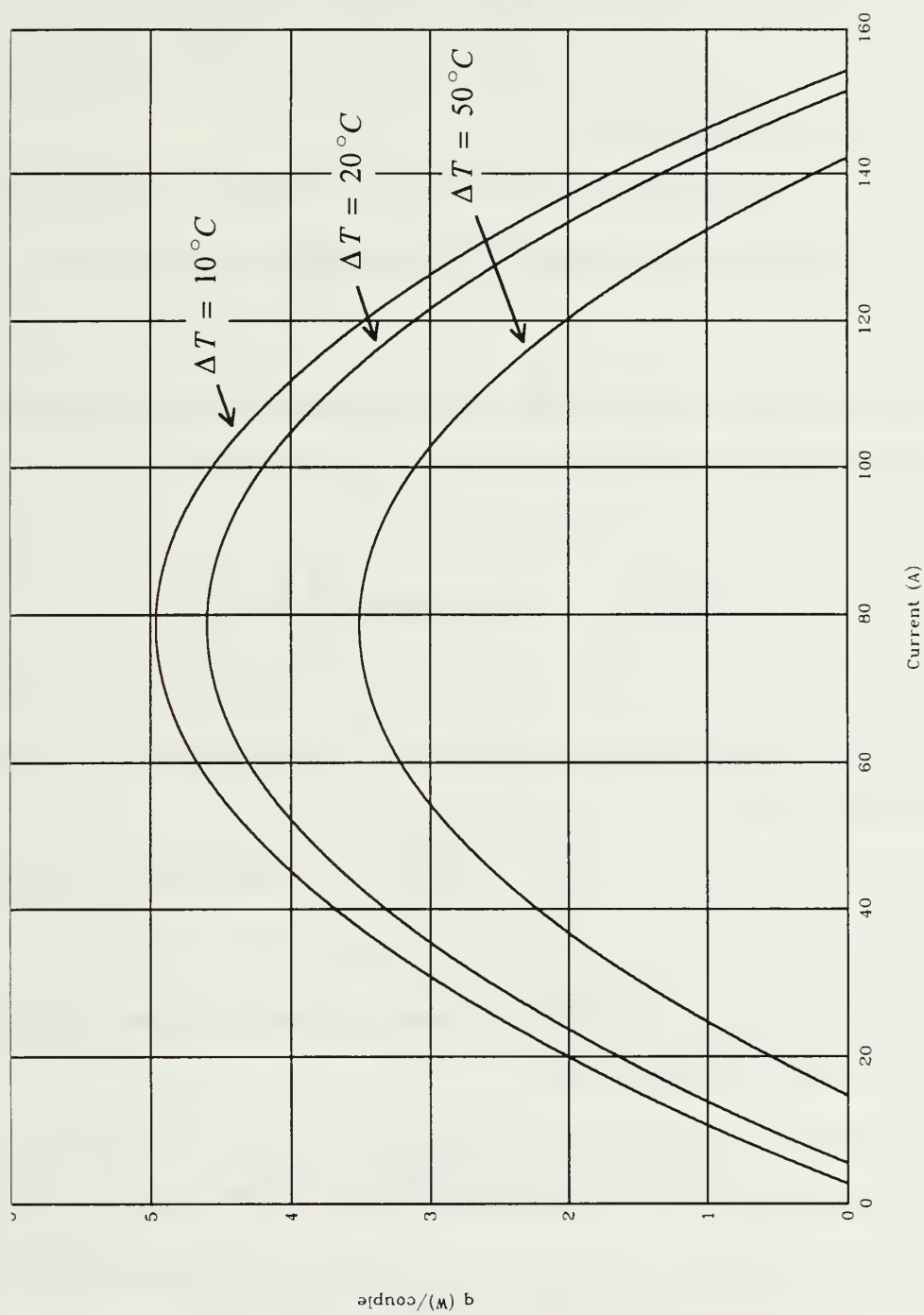


Figure 4 Heat Pumped as a Function of Current

$$KR = \left(\frac{\rho_A L_A}{A_A} + \frac{\rho_B L_B}{A_B} \right) \left(\frac{k_A A_A}{L_A} + \frac{k_B A_B}{L_B} \right) \quad (22)$$

and if $L_A = L_B$ an expansion gives

$$KR = k_A \rho_A + k_B \rho_A \frac{A_B}{A_A} + k_A \rho_B \frac{A_A}{A_B} + k_B \rho_B \quad (23)$$

An optimization of equation (23) can be accomplished by finding the point where the derivative with respect to the area ratio, A_A/A_B , vanishes:

$$\frac{dKR}{d(A_A/A_B)} = k_A \rho_B - k_B \rho_A \left(\frac{A_B}{A_A} \right)^2 = 0 \quad (24)$$

Then, a solution of equation (24) yields the area ratio that minimizes KR . Thus, in turn yields the optimal ΔT_m :

$$\frac{A_A}{A_B} = \sqrt{\frac{\rho_A k_B}{\rho_B k_A}} \quad (25)$$

The area ratio given by equation (25) when substituted into equation (22), yields the optimum value of KR , designated by ϕ

$$\phi = k_A \rho_A + 2 \sqrt{k_A \rho_A k_B \rho_B} + k_B \rho_B$$

or

$$\phi \equiv \left(\sqrt{k_A \rho_A} + \sqrt{k_B \rho_B} \right)^2 \quad (26)$$

Equation (16) has the restrictions that $L_A = L_B$ and that the area ratio be determined using equation (25).

E. FIGURE OF MERIT OF ELEMENT MATERIALS

Substituting ϕ , equation (26), into the equation for ΔT_m equation (21) :

$$\Delta T_m = \frac{\alpha^2}{2\phi} T_c^2$$

or

$$\Delta T_m = \frac{1}{2} z T_c^2$$

where z , the figure of merit of the materials, is defined as:

$$z \equiv \frac{\alpha^2}{\phi}$$

which may be rewritten as:

$$z = \frac{\alpha^2}{\left(\sqrt{k_A \rho_A} + \sqrt{k_B \rho_B} \right)^2}$$

When $L_A = L_B$ and the area ratio determined from equation (25), the figure of merit is a function of the properties of the elements. To maximize the figure of merit, the materials used should have a high Seebeck coefficient, low thermal conductivity, and low electrical resistivity Cadoff [Ref. 3: p. 21].

F. POWER REQUIRED TO OPERATE THE COUPLE

To operate the couple, a voltage equal to the sum of the Seebeck voltages in the couple and the resistive voltage drop must be applied Kraus [Ref. 1: p. 304]. The power required to operate the couple is then:

$$P = IV = I(\alpha \Delta T + IR) \quad (\text{W}) \quad (27)$$

G. COEFFICIENT OF PERFORMANCE

The coefficient of performance, η , measures the performance of the couple and is the heat pumped, q , divided by the power required to operate the couple, P :

$$\eta = \frac{q}{P} \quad (28)$$

Substituting equation (18) for q and equation (27) for P gives

$$\eta = \frac{\alpha I T_c - \frac{1}{2} I^2 R - K \Delta T}{I(IR + \alpha \Delta T)} \quad (29)$$

1. Coefficient Of Performance At Maximum Heat Pumping

One may express q_m in terms of ΔT and ΔT_m using equations (20) and (21):

$$q_m = \frac{\alpha^2 T_c^2}{2R} - K\Delta T = \frac{K\alpha^2 T_c^2}{2KR} - K\Delta T$$

$$q_m = K\Delta T_m - K\Delta T = K(\Delta T_m - \Delta T)$$

$$q_m = K\Delta T_m \left(1 - \frac{\Delta T}{\Delta T_m}\right) \quad (30)$$

In similar fashion, P_m may be expressed in terms of the temperatures by substituting equation (19) into equation (27):

$$P_m = \alpha I_m \Delta T + I_m^2 R$$

$$= \frac{\alpha^2 T_c \Delta T}{R} + \frac{\alpha^2 T_c^2}{R}$$

$$= \frac{\alpha^2 T_c^2}{R} \left(\frac{\Delta T}{T_c} + 1\right)$$

$$= \frac{2K\alpha^2 T_c^2}{2KR} \left(1 + \frac{\Delta T}{T_c}\right)$$

$$= 2K\Delta T_m \left(1 + \frac{\Delta T}{T_c}\right) \quad (31)$$

To obtain the coefficient of performance when the couple is adjusted for maximum heat pumping, substitute equations (30) and (31) into equation (28) for the coefficient of performance:

$$\eta_m = \frac{q_m}{P_m} = \frac{K\Delta T_m (1 - \Delta T/\Delta T_m)}{2K\Delta T_m(1 + \Delta T/\Delta T_c)}$$

$$\eta_m = \frac{1 - \Delta T/\Delta T_m}{2(1 + \Delta T/\Delta T_c)}$$

2. Current Yielding Optimum Coefficient Of Performance

The expression for the coefficient of performance, equation (29),

$$\eta = \frac{\alpha I T_c - \frac{1}{2}I^2 R - K\Delta T}{I(IR + \alpha \Delta T)}$$

may be differentiated to find the current that yields the maximum coefficient of performance. This current is the value of I that causes $d\eta/dI$ to vanish:

$$\frac{d\eta}{dI} = \frac{I(IR + \alpha \Delta T)(\alpha T_c - IR) - (\alpha I T_c - I^2 R/2 - K\Delta T)(2IR + \alpha \Delta T)}{I^2(IR + \alpha \Delta T)^2} = 0$$

or

$$(I^2 R + \alpha I \Delta T)(\alpha T_c - IR) = (\alpha I T_c - I^2 R/2 - K\Delta T)(2IR + \alpha \Delta T)$$

Expansion gives:

$$I^2 R \alpha T_c - I^3 R^2 + \alpha^2 I T_c \Delta T - \alpha I^2 R \Delta T = 2I^2 R \alpha T_c + \alpha^2 I T_c \Delta T - I^3 R^2 - \alpha I^2 R \Delta T/2 - 2IRK\Delta T - \alpha K\Delta T^2$$

or

$$\alpha I^2 R \Delta T/2 + \alpha T_c I^2 R - 2KIR\Delta T - K\alpha \Delta T^2 = 0$$

which is actually a quadratic in I :

$$R\alpha\left(\frac{\Delta T}{2} + T_c\right)I^2 - (2KR\Delta T)I - \left(K\alpha\Delta T^2\right) = 0$$

which possesses a solution

$$I = \frac{2KR\Delta T \pm \sqrt{(2KR\Delta T)^2 + 4KR\alpha^2\Delta T^2[(\Delta T/2) + T_c]}}{2R\alpha[(\Delta T/2) + T_c]} \quad (32)$$

The value of the radical in equation (32) will always be greater than $2KR\Delta T$ because it contains the sum of $(2KR\Delta T)^2$ and another, positive term. For cooling to occur, the current must not be negative so the solution in which the radical term is subtracted is ignored:

$$I = \frac{2KR\Delta T + \sqrt{(2KR\Delta T)^2 + 4KR\alpha^2\Delta T^2[(\Delta T/2) + T_c]}}{2R\alpha[(\Delta T/2) + T_c]}$$

factoring $4(KR)^2\Delta T^2$ from under the radical gives:

$$I = \frac{2KR\Delta T + 2KR\Delta T\sqrt{1 + (\alpha^2/KR)[(\Delta T/2) + T_c]}}{2R\alpha[(\Delta T/2) + T_c]}$$

or

$$I = KR\Delta T \frac{1 + \sqrt{1 + z[(\Delta T/2) + T_c]}}{R\alpha[(\Delta T/2) + T_c]}$$

Because

$$\frac{\Delta T}{2} + T_c = \frac{T_h - T_c}{2} + T_c = \frac{T_h + T_c}{2}$$

we define the average temperature may be defined as:

$$\bar{T} \equiv \frac{T_h + T_c}{2} \quad (33)$$

and

$$I = \frac{K\Delta T}{\alpha} \left\{ \frac{1 + \sqrt{1 + z[(T_h + T_c)/2]}}{(T_h + T_c)/2} \right\} \quad (34)$$

Multiplying equation (34) by

$$\frac{R\alpha}{R\alpha} \left(\frac{\sqrt{1 + z\bar{T}} - 1}{\sqrt{1 + z\bar{T}} - 1} \right) = 1$$

and substitution of the average temperature given by equation (33) one obtains:

$$I = \frac{KR\alpha\Delta T}{R\alpha^2} \left\{ \frac{1 + \sqrt{1 + z\bar{T}}}{\bar{T}} \right\} \left(\frac{\sqrt{1 + z\bar{T}} - 1}{\sqrt{1 + z\bar{T}} - 1} \right)$$

With $1/z = KR/\alpha^2$, expansion gives:

$$I = \frac{\alpha\Delta T}{R} \left(\frac{z\bar{T}}{z\bar{T}\sqrt{1 + z\bar{T}} - 1} \right)$$

Rearranging yields the expression for the current that optimizes the coefficient of performance:

$$I_o = \frac{\alpha \Delta T}{R \left(\sqrt{1 + z \bar{T}} - 1 \right)} \quad (35)$$

Figure 5 is a plot of the coefficient of performance as a function of current based on the conditions specified in equations (11) through (16).

3. Optimum Coefficient Of Performance

Substituting I_o into the coefficient of performance equation (35) yields the optimum coefficient of performance:

$$\eta_o = \frac{T_c}{\Delta T} \left[\frac{\sqrt{1 + z \bar{T}} - \frac{T_h}{T_c}}{\sqrt{1 + z \bar{T}} + 1} \right] \quad (36)$$

As z approaches ∞ , η_o approaches the ideal, or Carnot, coefficient of performance which is the term to the left of the brackets in equation (36) Egli [Ref. 4: p. 31] and Tipler [Ref. 5:pp 575-576].

H. CASCADED COOLERS

If a large ΔT is required, a multistage or cascaded cooler can be used. In a cascaded cooler the heat rejected by the first stage of couples is the heat load fed to the second stage. This arrangement can be extended to any number of stages. For best results when designing a multistage cooler for the lowest temperature attainable, each succeeding

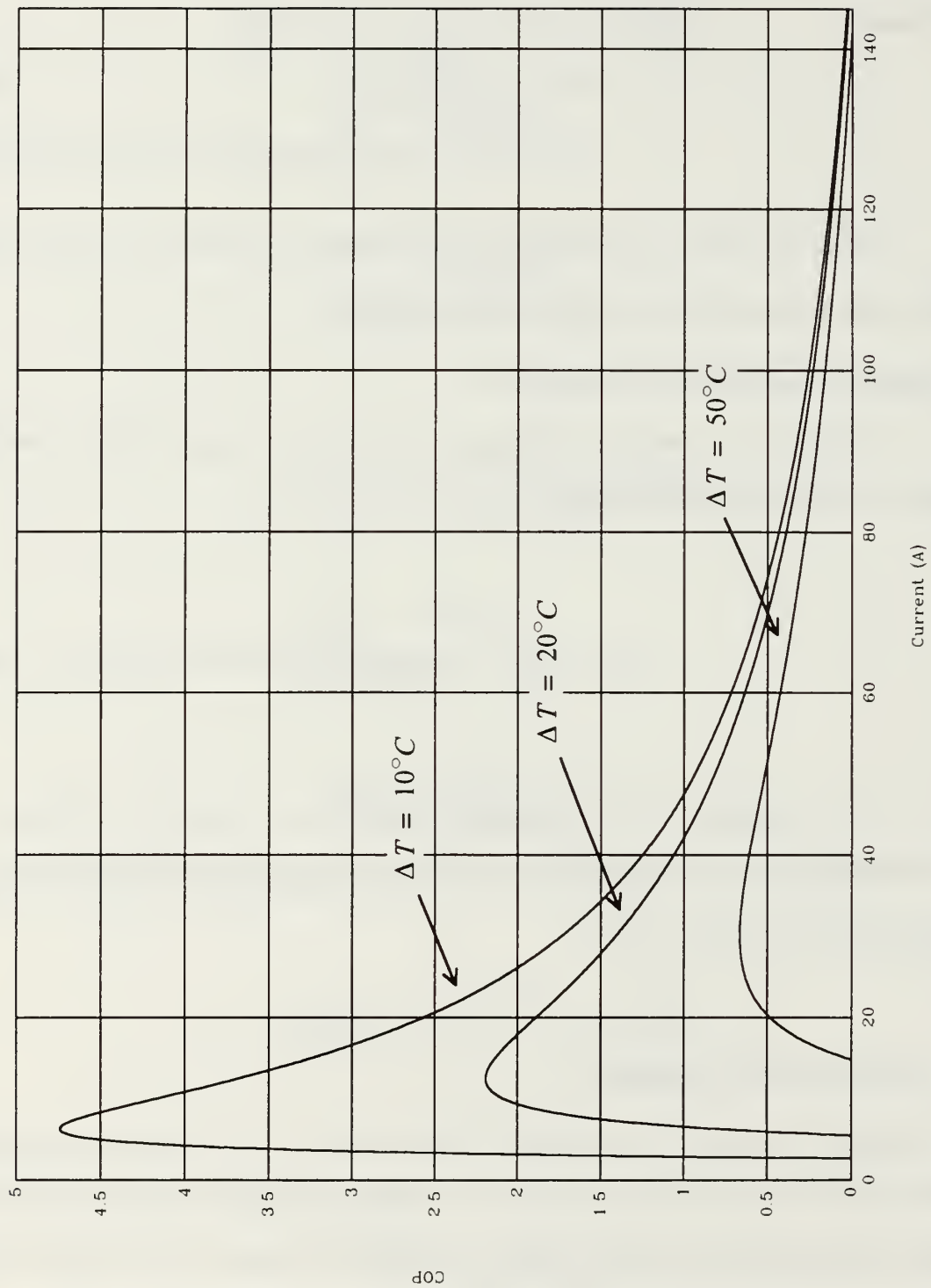


Figure 5 Coefficient of Performance as a Function of Current

stage should require less power than the previous stage. It is also possible to increase the coefficient of performance when the required temperature drop is close to the maximum for a given figure of merit Ioffe [Ref. 2: p. 116].

1. Overall Heat Pumped By Cascaded Coolers

The heat pumped by a battery of n -stages of cascaded coolers is equal to the summation of the heat load, q_1 at the first stage and the power required, P , of each stage over all stages in the battery:

$$q_{overall} = q_1 + \sum_{i=1}^N P_i \quad (\text{W})$$

2. Overall Coefficient Of Performance

The coefficient of performance of a battery n -stages of cascaded coolers is the heat load of the first stage, q_1 , divided by the total power required to operate the battery of coolers.

$$\eta_{overall} = \frac{q_1}{\sum_{i=1}^N P_i}$$

I. SUMMARY OF SIMPLIFYING ASSUMPTIONS

1. Thomson Voltage

Neglect of the Thomson voltage results in no significant error. However, designs utilizing this model will be conservative, understating couple performance Kraus [Ref. 1: p. 322].

2. Temperature Losses

Temperature losses between the thermoelectric couple and the heat load or heat exchanger are not taken into account directly. However, once determined, these temperature losses can be compensated for by adding the drop to the cold side temperature and subtracting the drop from the hot side temperature.

3. Parallel Thermal Paths

Insulation between couples is assumed to be perfect, eliminating parallel thermal path heat losses.

4. Junction Resistance

Junction resistance is assumed to be zero.

5. Figure Of Merit

In order to maximize the figure of merit, the couple material lengths are assumed to be equal and area ratio determined by equation (25).

6. Resistivity And Conductivity

Electrical resistivity, ρ , and thermal conductivity, k , are assumed to be constant with respect to temperature.

7. Couple Power Supply

The couple power supply is assumed to be direct current.

IV. THERMOELECTRIC COOLER DESIGN PROGRAM

A. INTRODUCTION

1. Features

Thermoelectric Cooler Design is a Microsoft Windows program to be used as an aid in the design of thermoelectric cooler devices. Use of the Microsoft Windows graphical user interface allows the user who is familiar with the Windows environment to concentrate on the design process instead of a new, nonstandard interface. This program was written to be used to quickly model and compare alternative designs. Modifications to a design can be easily incorporated. Major features of the program include:

1. Ability to change material properties and dimensions of couples.
2. Determine optimum coefficient of performance of each couple.
3. Determine maximum heat pumping capacity of each couple.
4. The ability to analyze cascaded couples.
5. Graph the performance parameters of couple.
6. Provide on-line help in the form of a glossary, diagrams, and a temperature conversion calculator.

2. System Requirements

The minimum software and hardware requirements needed by the computer system to run Thermoelectric Cooler Design are:

1. Microsoft Windows, version 3.0 or later.
2. A hard drive with 500 kilobytes of free space.
3. A mouse that is supported by Windows.

3. Disk Contents

Files included on the installation disk:

1. VBRUN100.DLL - Visual Basic kernel which must be in the same directory as THERMAL.EXE.
2. THERMAL.EXE - Thermoelectric Cooler Design executable file.
3. README.TXT - This manual in ASCII format.

Both VBRUN100.DLL and THERMAL.EXE must be installed in order to run Thermoelectric Cooler Design.

4. Installation

To install Thermoelectric Cooler Design:

1. Insert the installation disk in A: drive.
2. Exit to DOS. (Ensure C:\ prompt is present).
3. Type MD THERMAL then press the ENTER key.
4. Type COPY A:*. * C:\THERMAL then press the enter key. These commands create a new directory named THERMAL on the C drive and

copy the files from the installation disk into the C:\THERMAL directory. However, this is only one of many different ways to copy the files from the installation disk to the computer. If the choice is to install Thermoelectric Cooler Design on another drive or in another directory, both VBRUN100.DLL and THERMAL.EXE must be installed together in the same directory. The program will not run if this is not the case.

5. Run Microsoft Windows.
6. Select the Program Manager.
7. Open the Group Window in which Thermoelectric Cooler Design is to be installed.
8. Choose the option "new" from the file menu of the Windows Program Manager window. The new program object dialog box will then appear.
9. Select the program item and choose OK. The program item properties dialog box appears.
10. Type a description to appear under the program icon.
11. Select the browse button to display a list of files and directories.
12. Select THERMAL.EXE and choose OK.
13. Select OK to close the program item properties dialog box.
14. The Thermoelectric Cooler Design program is now installed.

B. USING THE PROGRAM

1. Starting And Exiting The Program

To run Thermoelectric Cooler Design, simply double-click the program icon. After displaying the introduction, the program menu bar will appear. This program menu bar, titled "Thermoelectric Cooler Design" provides access to all program functions. To increase readability, select the minimize on use option under the Windows program manager options menu.



Figure 6 Program Menu Bar

To exit the program, select exit from the program menu bar and choose yes when prompted. Once exited, all current design data is lost.

2. Setting Material Properties

Prior to determining the optimum coefficient of performance or maximum heat pumping capacity of a couple, the material properties and dimensions of the couple must be input.

To begin, select material properties from the program menu bar. The material properties menu will appear with default values. If these values are acceptable, choose OK and proceed to the optimize menu. If the default values are not acceptable they may be changed by:

1. Choosing either the first or second material. The α , ρ and k of the selected material is then displayed. All values in blue can be changed by the user. Appropriate values of α , ρ and k may be entered by the user by highlighting the text to be changed with the mouse and inputting the desired value from the keyboard.
2. Changes to the dimensions of the couple are input in a similar manner. Both the area and length of the second material may be input directly by the user.
3. Once the desired material properties and couple dimensions have been input, the overall resistance, thermal conductivity, and figure of merit are calculated by selecting the update button. In order to proceed to the optimize menu, select OK. To return all values to the defaults, choose the default button.

Material Properties

Properties
☒ First Material
☐ Second Material

α $V/^{\circ}C$
 ρ $\Omega \text{ cm}$
 k $W/cm^{\circ}C$

Dimensions

 First Material **Second Material**

Area cm^2 cm^2
Effective Diameter cm cm
Length cm cm

Overall

Resistance (R) Ω
Thermal Conductivity (K) $W/^{\circ}C$
Figure of Merit (z)

Figure 7 Material Properties Window

3. Optimum Coefficient Of Performance

To calculate the optimum coefficient of performance of a couple:

1. Ensure that the material properties menu has been reviewed.
2. Select the coefficient of performance option of the optimize menu.
3. The performance window will then appear. For a given low temperature the maximum high temperature is computed. The default ΔT is the maximum and may be changed by manipulating the vertical scroll bar. Once the heat load, low temperature and ΔT are acceptable, the optimum coefficient of performance is calculated by pressing the compute button. The optimum coefficient of performance and associated performance values are then displayed. Changes to the heat load, low temperature, and ΔT can be made and all couple performance values recomputed by choosing restart.

Performance

Heat Load (W): 300 455.2

High Temp (K): 455.1 Delta T: 146.5

Low Temp (K): 308.6 308.6

Buttons: Cancel, Compute

Vertical scroll bar on the right side of the input fields.

Figure 8 Optimum Coefficient of Performance Window

| Optimum Coefficient of Performance | | | | | | | | | |
|------------------------------------|--------|----|--------|---|----------------------------|--------|---|--|--|
| Print | | | | | | | | | |
| A/couple | 23.37 | qT | 300.0 | W | Number of Couples | 229 | | | |
| V/couple | 0.0569 | qP | 3.1739 | W | Total Heat Rejected | 604.8 | W | | |
| W/couple | 1.33 | qJ | 0.4696 | W | Coefficient of Performance | 0.9843 | | | |
| | | qF | 1.3941 | W | Power Consumption | 304.8 | W | | |
| | | qo | 1.3102 | W | | | | | |

Figure 9 Coefficient of Performance Results Window

4. Maximum Heat Pumping

To calculate the maximum heat pumping capacity of a couple:

1. Ensure that the material properties menu has been reviewed.
2. Select maximum the heat pumping option of the optimize menu.
3. The pumping window will appear. For a given low temperature the maximum high temperature is computed and displayed. The default temperature difference, ΔT , is the maximum and may be changed by manipulating the vertical scroll bar. Once the heat load, low temperature, and ΔT are acceptable, the maximum heat pumped is calculated by choosing compute. The maximum heat pumped and associated couple performance

values are then displayed. Changes to the heat load, low temperature and ΔT can be made and all couple performance values recomputed by choosing restart.

Figure 10 Maximum Heat Pumped Window

| Maximum Heat Pumping | | | | | | | | | |
|----------------------|--------|----|---------|---|----------------------------|---------|---|--|--|
| Print | | | | | | | | | |
| A/couple | 78.99 | qT | 300.0 | W | Number of Couples | 76 | | | |
| V/couple | 0.1525 | qP | 10.7253 | W | Total Heat Rejected | 1,215.8 | W | | |
| W/couple | 12.05 | qJ | 5.3626 | W | Coefficient of Performance | 0.3294 | | | |
| | | qF | 1.3941 | W | Power Consumption | 915.76 | W | | |
| | | qo | 3.9685 | W | | | | | |

Figure Eleven Heat Pumped Results Window

5. Cascading Stages

Once the initial couple performance values have been computed for either optimum coefficient of performance or maximum heat pumping, cascading of multiple couple stages can be simulated by selecting cascade. The cascade button is not visible

until the first stage couple performance has been calculated. Each additional stage assumes the rejected heat of the previous stage at the heat load and the high temperature of the previous stage. The maximum temperature at the new low temperature is then calculated. While the ΔT of the previous stage is assumed as a default, the ΔT for every stage of the cascade can be changed with the vertical scroll bar. While the program only displays five stages at a time, an unlimited number of cascade stages can be computed and displayed. Results of stages beyond five are shown in either the optimum coefficient of performance or maximum heat pumping windows, but not in the cascade window.

| Cascade | | | | | |
|-------------|--------|--------|--------|--------|--------|
| Print | | | | | |
| Stage | One | Two | Three | Four | Five |
| q in (W) | 300.0 | 1,215. | 4,626. | 16,850 | 59,511 |
| q out (W) | 1,215. | 4,626. | 16,850 | 59,511 | 205,37 |
| Couples | 76 | 227 | 667 | 1,943 | 5,629 |
| W/Couple | 12.049 | 15.024 | 18.326 | 21.956 | 25.912 |
| T cold (K) | 308.60 | 346.70 | 384.80 | 422.90 | 461.00 |
| T hot (K) | 346.70 | 384.80 | 422.90 | 461.00 | 499.10 |
| Delta T | 38.100 | 38.100 | 38.100 | 38.100 | 38.100 |
| Stage COP | 0.3294 | 0.3577 | 0.3789 | 0.3952 | 0.4080 |
| Overall COP | 0.3294 | 0.0697 | 0.0182 | 0.0051 | 0.0015 |

Figure Twelve Cascaded Stages Window

6. Graphing The Coefficient Of Performance And Heat Pumped

Thermoelectric Cooler Design is able to graph both coefficient of performance and heat pumped as a function of current. These graphs are useful in determining the

effect of a small change in current on both the maximum coefficient of performance and the heat pumped. The effects of varying the temperature differential across the couple can also be compared.

To graph either the coefficient of performance or heat pumped as a function of current:

1. Ensure that the material properties have been reviewed and optimum coefficient of performance or maximum heat pumped has been determined. The graph option of the program menu bar will not be available until these steps are accomplished.
2. Select the graph option from the program menu bar. The graph window will appear.
3. Select the function option from the graph window menu bar. Choose either coefficient of performance versus current or heat pumped versus current.
4. Choose draw from the graph window menu bar. The selected function is then graphed for the temperature differential, ΔT , chosen in the performance or heat pumped window.
5. To change the temperature differential, manipulate the ΔT vertical scroll bar in either the performance or heat pumped window as appropriate.

To superimpose plots of representative of various temperature differentials and functions, select superimpose from under the options choice of the graph window menu bar. For example, to compare the effect of varying the temperature differential on a graph, change the ΔT vertical scroll bar in either the performance or heat pumped window. Subsequent draw commands are then superimposed on the current graph. To clear the plot, deselect superimpose and then select draw

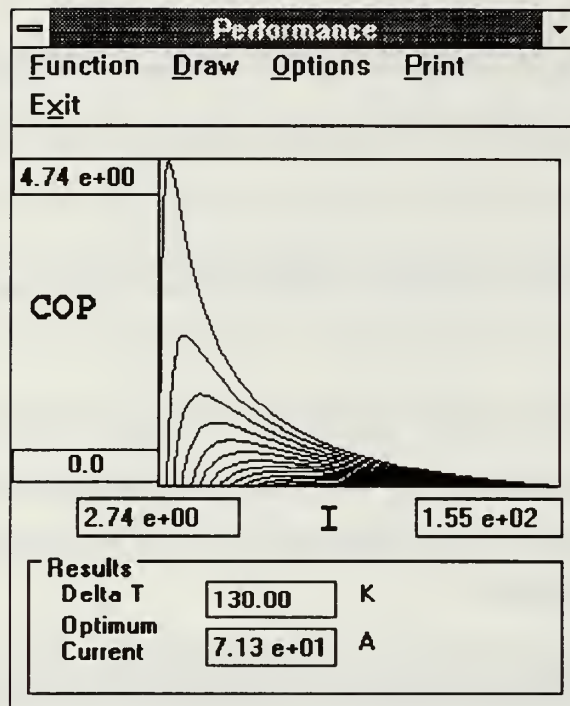


Figure Thirteen Optimum Coefficient of Performance Graph

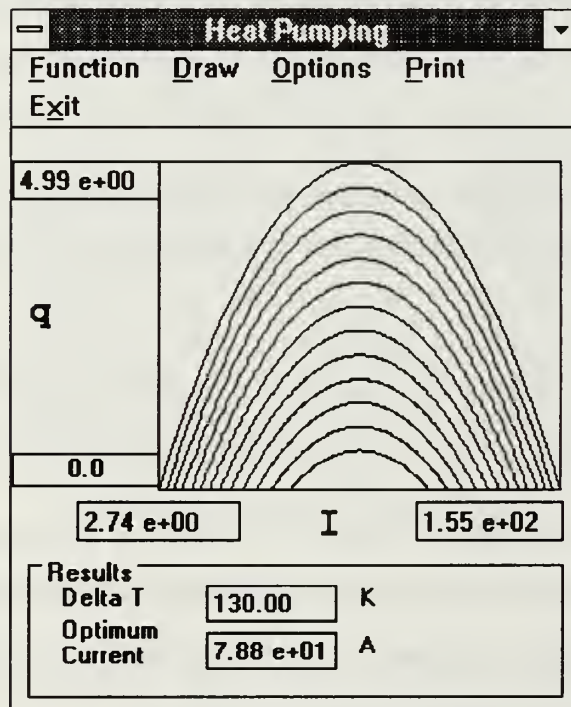


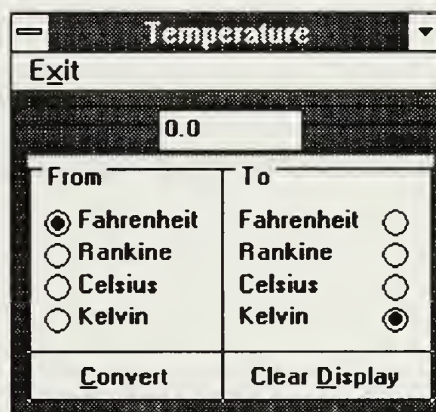
Figure Fourteen Maximum Heat Pumped Graph

The number of samples used to produce the graph may be adjusted as required. Select the sample size desired under the samples choice under the options selection of the graph window menu bar.

7. Temperature Conversion Calculator

To convert between temperature scales:

1. Select the calculator option of the program menu bar. The temperature conversion calculator will appear.
2. Input the temperature to convert from by highlighting the blue numerical display with the mouse by holding down the left mouse button while sweeping the cursor from left to right across the calculator display text. Once the text is highlighted, type in the desired temperature.
3. Select the temperature scale to convert "from".
4. Select the temperature scale to convert "to".
5. Choose convert and read the temperature in the calculator display.
6. To clear the display, select clear display option.



The screenshot shows a window titled "Temperature" with a standard Mac OS-style title bar (minimize, maximize, close buttons). Below the title bar is a menu bar with the "Exit" option. The main area of the window contains a numerical display showing "0.0". Below the display are two columns of radio buttons for selecting temperature scales. The "From" column has four options: Fahrenheit (selected), Rankine, Celsius, and Kelvin. The "To" column also has four options: Fahrenheit, Rankine, Celsius, and Kelvin (selected). At the bottom of the window are two buttons: "Convert" and "Clear Display".

| From | To |
|---|---|
| <input checked="" type="radio"/> Fahrenheit | Fahrenheit <input type="radio"/> |
| <input type="radio"/> Rankine | Rankine <input type="radio"/> |
| <input type="radio"/> Celsius | Celsius <input type="radio"/> |
| <input type="radio"/> Kelvin | Kelvin <input checked="" type="radio"/> |

Convert Clear Display

Figure Fifteen Temperature Conversion Calculator

8. Diagrams And Glossary

On-line assistance is available in the form of diagrams and a glossary.

Diagrams of a single couple and cascaded couples are available by selecting the diagram option from the program menu bar.

A glossary of terms used throughout the program is available. Choose the glossary option of the program menu bar.

V. CONCLUSIONS

The biggest drawback of a thermoelectric cooler is its inherent low coefficient of performance, but new materials with better thermoelectric properties could eventually make thermoelectric devices competitive with compressor systems. While unable to compete directly with compressor based systems, thermoelectric coolers do have many unique characteristics making them useful in a variety of applications. For spaceborne applications, high reliability, vibration free operation, and no dependence on consumables are their most important features.

The design of thermoelectric coolers is tedious without the aid of a computer. Unfortunately, the utility of many programs written for DOS are handicapped by a poor user interface. More time is often required to figure out how to use the program than to solve the initial problem. In writing thermoelectric cooler design, the Windows graphical user interface was used to provide a better user interface. Windows offers the programmer several powerful program development tools. For the user, Windows offers a familiar working environment enabling the user to concentrate on the task at hand.

APPENDIX

Computer code listing for Thermoelectric Cooler Design program.

Thesis.txt

```
// Code for main menu bar
```

```
Sub ProgramInfoName_Click ()
```

```
    MsgBox "LCDR Clifton's Naval Postgraduate School Thesis"
```

```
End Sub
```

```
Sub DefinitionsName_Click ()
```

```
    Definitions.Show
```

```
End Sub
```

```
Sub TemperatureName_Click ()
```

```
    Screen.MousePointer = 11
```

```
    TempCalc.Show
```

```
    TempCalc.WindowState = 0
```

```
    Screen.MousePointer = 0
```

```
End Sub
```

```
Sub ChooseName_Click ()
```

```
    Material.Show
```

```
End Sub
```

```
Sub ShowGraphName_Click ()
```

```
    Screen.MousePointer = 11
```

```
    Graph.Show
```

```
    Graph.WindowState = 0
```

```
    Screen.MousePointer = 0
```

```
End Sub
```

```
Sub ExitName_Click ()
```

```
    Screen.MousePointer = 11
```

```
    Ending.Show
```

```
    Screen.MousePointer = 0
```

```
End Sub
```

```
Sub Form_Load ()
```

```
    Main.Left = (Screen.Width - Main.Width) / 2
```

```
    Main.Top = 0
```

```
End Sub
```



```

Sub Form_Click ()
    Main.Left = (Screen.Width - Main.Width) / 2
    Main.Top = 0

End Sub

Sub PerformanceName_Click ()
End Sub

Sub PumpingName_Click ()
    Pumping.Show
End Sub

Sub Command1_Click ()
    Results.CoeffText.Text = ""
    Results.NumberCouplesText.Text = ""
    Results.TotalHeatRejectedText.Text = ""
    Results.AperCoupleText.Text = ""
    Results.VperCoupleText.Text = ""
    Results.WperCoupleText.Text = ""
    Results.qTText.Text = ""
    Results.qPText.Text = ""
    Results.qJText.Text = ""
    Results.qFText.Text = ""
    Results.qoText.Text = ""

End Sub

Sub InfoName_Click ()
    MsgBox "LCDR Clifton's Naval Postgraduate School Thesis"
End Sub

Sub SinglePumpingName_Click ()

    Screen.MousePointer = 11

    ScrollUpdate = False

    Tc = Val(Pumping.LowTempText.Text)
    DeltaTMax = .5 * z * (Tc ^ 2)
    TMax = DeltaTMax + Tc

    Pumping.Tempscroll.Max = Tc * 10 + 1

```

```

Pumping.Tempscroll.Min = TMax * 10 - 1
Pumping.Tempscroll.Value = Pumping.Tempscroll.Min

Pumping.TMinDisplay.Text = Format$(Tc, "##,##0.0")
Pumping.TMaxDisplay.Text = Format$(DeltaTMax + Tc, "##,##0.0")

Th = Pumping.Tempscroll.Value / 10
Pumping.HighTempDisplay.Text = Format$(Th, "##,##0.0")
DeltaT = Th - Tc
Pumping.DeltaTDisplay.Text = Format$(DeltaT, "##,##0.0")

Pumping.Show
Pumping.WindowState = 0

Screen.MousePointer = 0

ScrollUpdate = True

End Sub

Sub GlossaryName_Click ()
    Screen.MousePointer = 11
    Definitions.Show
    Definitions.WindowState = 0
    Screen.MousePointer = 0
End Sub

Sub CoolerName_Click ()
    Screen.MousePointer = 11
    Cooler.Show
    Cooler.WindowState = 0
    Screen.MousePointer = 0
End Sub

Sub CascadeName_Click ()
    Screen.MousePointer = 11
    Cascade.Show
    Cascade.WindowState = 0
    Screen.MousePointer = 0
End Sub

Sub SinglePerformanceName_Click ()

    Screen.MousePointer = 11

```

ScrollUpdate = False

Tc = Val(Performance.LowTempText.Text)

DeltaTMax = .5 * z * (Tc ^ 2)

TMax = DeltaTMax + Tc

Performance.Tempscroll.Max = Tc * 10 + 1

Performance.Tempscroll.Min = TMax * 10 - 1

Performance.Tempscroll.Value = Performance.Tempscroll.Min

Performance.TMinDisplay.Text = Format\$(Tc, "##,##0.0")

Performance.TMaxDisplay.Text = Format\$(TMax, "##,##0.0")

Th = Performance.Tempscroll.Value / 10

Performance.HighTempDisplay.Text = Format\$(Th, "##,##0.0")

DeltaT = Th - Tc

Performance.DeltaTDisplay.Text = Format\$(DeltaT, "##,##0.0")

Performance.Show

Performance.WindowState = 0

Screen.MousePointer = 0

ScrollUpdate = True

End Sub

Sub MaterialName_Click ()

Screen.MousePointer = 11

alpha_A = .00023

alpha_B = .00021

rho_A = .001!

rho_B = .001!

kappa_A = .017

kappa_B = .0145

Aa_over_Ab = Sqr((kappa_B * rho_A) / (kappa_A * rho_B))

Ab = .3848

Aa = Aa_over_Ab * Ab

dB = 2 * Sqr(Ab / pi)

dA = 2 * Sqr(Aa / pi)

```

La = .318
Lb = La
alpha = alpha_A + alpha_B

z = (alpha ^ 2) / ((Sqr(kappa_A * rho_A) + Sqr(kappa_B * rho_B)) ^ 2)
R = (rho_A * La / Aa) + (rho_B * Lb / Ab)
K = ((kappa_A * Aa) / La) + ((kappa_B * Ab) / Lb)

Material.Option1.Value = True

Material.AlphaText.Text = Format$(alpha_A, "0.00000")
Material.RhoText.Text = Format$(rho_A, "0.00000")
Material.kText.Text = Format$(kappa_A, "0.00000")

Material.FirstMaterial_a_Display.Text = Format$(Aa, "##0.0000")
Material.FirstMaterial_d_Display.Text = Format$(dA, "##0.0000")
Material.SecondMaterial_d_Display.Text = Format$(dB, "##0.0000")
Material.FirstMaterial_L_Display.Text = Format$(La, "##0.0000")
Material.RDisplay.Text = Format$(R, "0.000E+00")
Material.KDisplay.Text = Format$(K, "0.000E+00")
Material.zDisplay.Text = Format$(z, "0.000E+00")

Material.Show
Material.WindowState = 0
Screen.MousePointer = 0
End Sub

Material.txt
// Code for Material Properties Window
Dim Aa, Ab, La, Lb As Double

Sub Form_Load ()

    Material.Left = (Screen.Width - Material.Width) / 2
    Material.Top = (Screen.Height - Material.Height) / 2
End Sub

Sub Materials_Click ()
End Sub

Sub Option1_Click ()
    If Option1.Value = True Then

```

```

AlphaText.Text = Format$(alpha_A, "0.00000")
RhoText.Text = Format$(rho_A, "0.00000")
kText.Text = Format$(kappa_A, "0.00000")
Else
AlphaText.Text = Format$(alpha_B, "0.00000")
RhoText.Text = Format$(rho_B, "0.00000")
kText.Text = Format$(kappa_B, "0.00000")
End If
End Sub

```

```

Sub Option2_Click ()
If Option1.Value = True Then
AlphaText.Text = Format$(alpha_A, "0.00000")
RhoText.Text = Format$(rho_A, "0.00000")
kText.Text = Format$(kappa_A, "0.00000")
Else
AlphaText.Text = Format$(alpha_B, "0.00000")
RhoText.Text = Format$(rho_B, "0.00000")
kText.Text = Format$(kappa_B, "0.00000")
End If
End Sub

```

```

Sub AlphaText_Change ()
If Option1.Value = True Then
alpha_A = Val(AlphaText.Text)
Else
alpha_B = Val(AlphaText.Text)
End If
End Sub

```

```

Sub RhoText_Change ()
If Option1.Value = True Then
rho_A = Val(RhoText.Text)
Else
rho_B = Val(RhoText.Text)
End If
End Sub

```

```

Sub kText_Change ()
If Option1.Value = True Then
kappa_A = Val(kText.Text)
Else
kappa_B = Val(kText.Text)

```

```

End If
End Sub

Sub Command1_Click ()
    Main.OptimizeName.Enabled = True
    Material.Hide
End Sub

Sub Command2_Click ()
    alpha_A = .00023
    alpha_B = .00021
    rho_A = .001!
    rho_B = .001!
    kappa_A = .017
    kappa_B = .0145

    Aa_over_Ab = Sqr((kappa_B * rho_A) / (kappa_A * rho_B))
    Ab = .3848
    Aa = Aa_over_Ab * Ab
    dB = 2 * Sqr(Ab / pi)
    dA = 2 * Sqr(Aa / pi)
    La = .318
    Lb = La
    alpha = alpha_A + alpha_B

    z = (alpha ^ 2) / ((Sqr(kappa_A * rho_A) + Sqr(kappa_B * rho_B)) ^ 2)
    R = (rho_A * La / Aa) + (rho_B * Lb / Ab)
    K = ((kappa_A * Aa) / La) + ((kappa_B * Ab) / Lb)

    Option1.Value = True

    AlphaText.Text = Format$(alpha_A, "0.00000")
    RhoText.Text = Format$(rho_A, "0.00000")
    kText.Text = Format$(kappa_A, "0.00000")

    FirstMaterial_a_Display.Text = Format$(Aa, "##0.0000")
    SecondMaterial_a_Display.Text = Format$(Ab, "##0.0000")
    FirstMaterial_d_Display.Text = Format$(dA, "##0.0000")
    SecondMaterial_d_Display.Text = Format$(dB, "##0.0000")
    FirstMaterial_L_Display.Text = Format$(La, "##0.0000")
    SecondMaterial_L_Display.Text = Format$(Lb, "##0.0000")
    RDisplay.Text = Format$(R, "0.000E+00")
    KDisplay.Text = Format$(K, "0.000E+00")
    zDisplay.Text = Format$(z, "0.000E+00")

```

End Sub

Sub Command3_Click ()

Aa_over_Ab = Sqr((kappa_B * rho_A) / (kappa_A * rho_B))

Ab = Val(SecondMaterial_a_Display.Text)

Aa = Aa_over_Ab * Ab

dB = 2 * Sqr(Ab / pi)

dA = 2 * Sqr(Aa / pi)

Lb = Val(SecondMaterial_L_Display.Text)

La = Lb

alpha = alpha_A + alpha_B

$z = (\alpha^2) / ((\text{Sqr}(\kappa_A * \rho_A) + \text{Sqr}(\kappa_B * \rho_B))^2)$

$R = (\rho_A * La / Aa) + (\rho_B * Lb / Ab)$

$K = ((\kappa_A * Aa) / La) + ((\kappa_B * Ab) / Lb)$

FirstMaterial_a_Display.Text = Format\$(Aa, "##0.0000")

SecondMaterial_a_Display.Text = Format\$(Ab, "##0.0000")

FirstMaterial_d_Display.Text = Format\$(dA, "##0.0000")

SecondMaterial_d_Display.Text = Format\$(dB, "##0.0000")

FirstMaterial_L_Display.Text = Format\$(La, "##0.0000")

SecondMaterial_L_Display.Text = Format\$(Lb, "##0.0000")

RDisplay.Text = Format\$(R, "0.000E+00")

KDisplay.Text = Format\$(K, "0.000E+00")

zDisplay.Text = Format\$(z, "0.000E+00")

End Sub

Coinput.txt

Sub Command1_Click ()

Screen.MousePointer = 11

Stage = 0

Performance.Hide

History.Hide

Results.Hide

Screen.MousePointer = 0

End Sub

Sub Form_Load ()

PerformanceCmd.Width = 2415

Performance.Left = Screen.Width - Performance.Width


```

Performance.Top = Main.Height
LowTempUpdate = True
Cascading = False

```

```

End Sub

```

```

Sub Command2_Click ()

```

```

    Screen.MousePointer = 11
    Command2.Caption = "Wait"
    History.WindowState = 0
    LowTempUpdate = True
    Cascading = True

```

```

    Stage = Stage + 1

```

```

    If Stage = 1 Then
        TotalPower = 0
    End If

```

```

        Tc = Val(LowTempText.Text)
        Th = Val(HighTempDisplay.Text)

```

```

        DeltaT = Th - Tc
        Th_plus_Tc = Th + Tc
        gamma = Sqr(1 + (z / 2) * (Th_plus_Tc))
        Io = (alpha * DeltaT) / (R * (gamma - 1))
        Vo = (alpha * DeltaT) + Io * R
        P = Io * Vo
        qT = Val(ReferLoadText.Text)
        qP = alpha * Io * Tc
        qJ = .5 * (Io ^ 2) * R
        qF = K * DeltaT
        qo = qP - (qJ + qF)
        No_of_Couples = qT / qo
        Total_Heat_rejected = qT + No_of_Couples * P
        Coeff_of_Performance = qo / P
        TotalPower = TotalPower + (P * No_of_Couples)
        If Stage = 1 Then
            FirstStage_qo = qo * No_of_Couples
        End If
        OverallCOP = FirstStage_qo / TotalPower

```

```

        TMaxDisplay.Text = Format$(DeltaTMax + Tc, "##,##0.0")

```

```

TMinDisplay.Text = Format$(Tc, "##,##0.0")
DeltaTDisplay.Text = Format$(DeltaT, "##,##0.0")

Results.CoeffText.Text = Format$(Coeff_of_Performance, "0.0000")
Results.NumberCouplesText.Text = Format$(No_of_Couples, "#,##0.00")
Results.TotalHeatRejectedText.Text = Format$(Total_Heat_rejected, "#,##0.0")
Results.AperCoupleText.Text = Format$(Io, "#,##0.00")
Results.VperCoupleText.Text = Format$(Vo, "#,##0.0000")
Results.WperCoupleText.Text = Format$(P, "#,##0.0000")
Results.qTText.Text = Format$(qT, "#,##0.0")
Results.qPText.Text = Format$(qP, "#,##0.0000")
Results.qJText.Text = Format$(qJ, "#,##0.0000")
Results.qFText.Text = Format$(qF, "#,##0.0000")
Results.qoText.Text = Format$(qo, "#,##0.0000")
Results.Caption = "Optimum Coefficient of Performance" + " Stage " + Str$(Stage)
Results.Show

```

Select Case Stage

Case 1

```

History.qLoadStageOneDisplay.Text = Format$(qT, "#,##0.0")
History.qRejectedStageOneDisplay.Text = Format$(Total_Heat_rejected,
"#,##0.0")
History.CouplesStageOneDisplay.Text = Format$(No_of_Couples,
"#,##0.00")
History.WperCoupleStageOneDisplay.Text = Format$(P, "#,##0.0000")
History.TcStageOneDisplay.Text = Format$(Tc, "0.0000")
History.ThStageOneDisplay.Text = Format$(Th, "0.0000")
History.DeltaTStageOneDisplay.Text = Format$(DeltaT, "0.0000")
History.StageCOPStageOneDisplay.Text = Format$(Coeff_of_Performance,
"0.0000")
History.OverallCOPStageOneDisplay.Text = Format$(OverallCOP, "0.0000")
History.Width = 2250

```

Case 2

```

History.qLoadStageTwoDisplay.Text = Format$(qT, "#,##0.0")
History.qRejectedStageTwoDisplay.Text = Format$(Total_Heat_rejected,
"#,##0.0")
History.CouplesStageTwoDisplay.Text = Format$(No_of_Couples,
"#,##0.00")
History.WperCoupleStageTwoDisplay.Text = Format$(P, "#,##0.0000")
History.TcStageTwoDisplay.Text = Format$(Tc, "0.0000")
History.ThStageTwoDisplay.Text = Format$(Th, "0.0000")
History.DeltaTStageTwoDisplay.Text = Format$(DeltaT, "0.0000")
History.StageCOPStageTwoDisplay.Text = Format$(Coeff_of_Performance,
"0.0000")

```

History.OverallCOPStageTwoDisplay.Text = Format\$(OverallCOP, "0.0000")
History.Width = 3360

Case 3

History.qLoadStageThreeDisplay.Text = Format\$(qT, "#,##0.0")
History.qRejectedStageThreeDisplay.Text = Format\$(Total_Heat_rejected,
"#,##0.0")
History.CouplesStageThreeDisplay.Text = Format\$(No_of_Couples,
"#,##0.00")
History.WperCoupleStageThreeDisplay.Text = Format\$(P, "#,##0.0000")
History.TcStageThreeDisplay.Text = Format\$(Tc, "0.0000")
History.ThStageThreeDisplay.Text = Format\$(Th, "0.0000")
History.DeltaTStageThreeDisplay.Text = Format\$(DeltaT, "0.0000")
History.StageCOPStageThreeDisplay.Text = Format\$(Coeff_of_Performance,
"0.0000")
History.OverallCOPStageThreeDisplay.Text = Format\$(OverallCOP,
"0.0000")
History.Width = 4425

Case 4

History.qLoadStageFourDisplay.Text = Format\$(qT, "#,##0.0")
History.qRejectedStageFourDisplay.Text = Format\$(Total_Heat_rejected,
"#,##0.0")
History.CouplesStageFourDisplay.Text = Format\$(No_of_Couples,
"#,##0.00")
History.WperCoupleStageFourDisplay.Text = Format\$(P, "#,##0.0000")
History.TcStageFourDisplay.Text = Format\$(Tc, "0.0000")
History.ThStageFourDisplay.Text = Format\$(Th, "0.0000")
History.DeltaTStageFourDisplay.Text = Format\$(DeltaT, "0.0000")
History.StageCOPStageFourDisplay.Text = Format\$(Coeff_of_Performance,
"0.0000")
History.OverallCOPStageFourDisplay.Text = Format\$(OverallCOP, "0.0000")
History.Width = 5520

Case 5

History.qLoadStageFiveDisplay.Text = Format\$(qT, "#,##0.0")
History.qRejectedStageFiveDisplay.Text = Format\$(Total_Heat_rejected,
"#,##0.0")
History.CouplesStageFiveDisplay.Text = Format\$(No_of_Couples,
"#,##0.00")
History.WperCoupleStageFiveDisplay.Text = Format\$(P, "#,##0.0000")
History.TcStageFiveDisplay.Text = Format\$(Tc, "0.0000")
History.ThStageFiveDisplay.Text = Format\$(Th, "0.0000")
History.DeltaTStageFiveDisplay.Text = Format\$(DeltaT, "0.0000")
History.StageCOPStageFiveDisplay.Text = Format\$(Coeff_of_Performance,
"0.0000")
History.OverallCOPStageFiveDisplay.Text = Format\$(OverallCOP, "0.0000")

History.Width = 6585

End Select

History.Show 0

ScrollUpdate = False

LowTempText.Text = Format\$(Th, "##,##0.0")

ScrollUpdate = True

ReferLoadText.Text = Format\$(Total_Heat_rejected, "##0.0")

LowTempUpdate = False

Tempscroll.Value = (Tc + DeltaT) * 10

LowTempUpdate = True

Command2.Caption = "Cascade"

Screen.MousePointer = 0

Cascading = False

End Sub

Sub TempScroll_Change ()

If ScrollUpdate = True Then

Tc = Val(LowTempText.Text)

Th = Tempscroll.Value / 10

DeltaT = Th - Tc

HighTempDisplay.Text = Format\$(Th, "##,##00.0")

DeltaTDisplay.Text = Format\$(DeltaT, "##,##00.0")

End If

End Sub

Sub LowTempText_Change ()

If LowTempUpdate = True Then

Tc = Val(LowTempText.Text)

DeltaTMax = .5 * z * (Tc ^ 2)

TMax = DeitaTMax + Tc

Tempscroll.Max = Tc * 10 + 1

Tempscroll.Min = TMax * 10 - 1

If Stage = 0 Then

Tempscroll.Value = Tempscroll.Min

End If

Th = Tempscroll.Value / 10

```

If Cascading = False Then
    DeltaT = Th - Tc
End If

TMinDisplay.Text = Format$(Tc, "##,##0.0")
TMaxDisplay.Text = Format$(TMax, "##,##0.0")
DeltaTDisplay.Text = Format$(DeltaT, "##,##0.0")
End If
End Sub

Sub PerformanceCmd_Click ()

    Screen.MousePointer = 11
    Main.ShowGraphName.Enabled = True
    PerformanceCmd.Caption = "Wait"
    Command2.Visible = -1
    History.Hide

    Stage = 0

    DeltaT = Th - Tc
    Th_plus_Tc = Th + Tc
    gamma = Sqr(1 + (z / 2) * (Th_plus_Tc))
    Io = (alpha * DeltaT) / (R * (gamma - 1))
    Vo = (alpha * DeltaT) + Io * R
    P = Io * Vo
    qT = Val(ReferLoadText.Text)
    qP = alpha * Io * Tc
    qJ = .5 * (Io ^ 2) * R
    qF = K * DeltaT
    qo = qP - (qJ + qF)
    No_of_Couples = qT / qo
    Total_Heat_rejected = qT + No_of_Couples * P
    Coeff_of_Performance = qo / P

    TMaxDisplay.Text = Format$(DeltaTMax + Tc, "##,##0.0")
    TMinDisplay.Text = Format$(Tc, "##,##0.0")
    DeltaTDisplay.Text = Format$(DeltaT, "##,##0.0")

    Results.CoeffText.Text = Format$(Coeff_of_Performance, "0.0000")
    Results.NumberCouplesText.Text = Format$(No_of_Couples, "#,##0.00")
    Results.TotalHeatRejectedText.Text = Format$(Total_Heat_rejected, "#,##0.0")
    Results.AperCoupleText.Text = Format$(Io, "#,##0.00")

```



```

Results.VperCoupleText.Text = Format$(Vo, "###0.0000")
Results.WperCoupleText.Text = Format$(P, "###0.00")
Results.qTText.Text = Format$(qT, "###0.0")
Results.qPText.Text = Format$(qP, "###0.0000")
Results.qJText.Text = Format$(qJ, "###0.0000")
Results.qFText.Text = Format$(qF, "###0.0000")
Results.qoText.Text = Format$(qo, "###0.0000")
Results.Caption = "Optimum Coefficient of Performance"
Results.Show
Results.PowerConsumptionDisplay.Text = Format$((P * No_of_Couples),
"#,###,###0.0")
PerformanceCmd.Width = 975
PerformanceCmd.Caption = "Restart"
Screen.MousePointer = 0
End Sub

```

Cooler.txt

```

// Code for Optimum Coefficient of Performance Window
DefDb1 A-Z

```

```
Global Const True = -1
```

```
Global Const False = 0
```

```
Global Const pi = 355 / 113
```

```
Global DeltaT As Double
```

```
Global DeltaTMax As Double
```

```
Global alpha_A, alpha_B, rho_A, rho_B, kappa_A, kappa_B As Double
```

```
Global alpha As Double
```

```
Global z, K, R As Double
```

```
Global Stage As Integer
```

```
Global ScrollUpdate As Integer
```

```
Global LowTempUpdate As Integer
```

```
Global Cascading As Integer
```

```
Global Total_Heat_rejected As Double
```

```
Global Tc As Double
```

```
Global Th As Double
```

```
Global TotalPower As Double
```

```
Global FirstStage_qo As Double
```

Cascade.txt

```

// Code for Cascaded Stages Window

```

```
Sub PrintName_Click ()
```

```
    Cascade.PrintForm
```

End Sub

```
Sub ExitName_Click ()  
    Cascade.Hide  
End Sub
```

Define.txt

```
// Code to call Definitions Window  
Sub PrintName_Click ()  
    Definitions.PrintForm  
End Sub
```

```
Sub ExitName_Click ()  
    Definitions.Hide  
End Sub
```

Diagram.txt

```
// Code to call Diagrams  
Sub PrintName_Click ()  
    Cooler.PrintForm  
End Sub
```

```
Sub ExitName_Click ()  
    Cooler.Hide  
End Sub
```

Ending.txt

```
// Code to end design session, closing all open windows  
Sub Command1_Click ()  
    End  
End Sub
```

```
Sub Command2_Click ()  
    Ending.Hide  
End Sub
```

```
Sub Form_Click ()  
    Main.Left = (Screen.Width - Main.Width) / 2  
    Main.Top = (Screen.Height - Main.Height) / 2
```

End Sub

```
Sub Form_Load ()
```



```
Ending.Left = (Screen.Width - Ending.Width) / 2
Ending.Top = (Screen.Height - Ending.Height) / 2
```

```
End Sub
```

Graph.txt

```
// Code for graping routines
Dim Shown As Integer
Dim Samples As Integer
Dim IMax As Double
Dim IMin As Double
Dim YMax As Double
Dim YMin As Double
Dim OldYMax As Double
Dim OldIMax As Double
Dim OldIMin As Double
Dim OldMaxCount As Double
Dim OldMinCount As Double
Dim IatOldMinCount As Double
Dim IatOldMaxCount As Double
Dim PerformancePlot As Integer
Dim SuperimposePlots As Integer
```

```
Sub GraphIt ()
```

```
    ReDim I(Samples)
    ReDim Y(Samples)
    IMin = ((2 * alpha * Tc) - Sqr((2 * alpha * Tc) ^ 2 - (8 * R * K * DeltaT))) / (2 * R)
    IMax = ((2 * alpha * Tc) + Sqr((2 * alpha * Tc) ^ 2 - (8 * R * K * DeltaT))) / (2 * R)
    YMin = 0
    YMax = 0
    MinCount = 0
    MaxCount = 0
    Minimum = False
```

```
    StepInterval = (IMax - IMin) / (Samples - 1)
```

```
    If PerformancePlot = True Then
        YAxisLabel.Text = "COP"
        Graph.Caption = "Performance"
    Else
        YAxisLabel.Text = "q"
        Graph.Caption = "Heat Pumping"
```

End If

For J = 1 To Samples

I(J) = IMin + StepInterval * (J - 1)

If PerformancePlot = True Then

Y(J) = ((alpha * I(J) * Tc) - (.5 * I(J) ^ 2 * R) - (K * DeltaT)) / (I(J) * ((alpha * DeltaT) + (I(J) * R)))

Else

Y(J) = (alpha * Tc * I(J)) - (.5 * I(J) ^ 2 * R) - (K * DeltaT)

End If

If Y(J) > 0 And Minimum = False Then

YMin = Y(J)

MinCount = J

Minimum = True

ElseIf Y(J) > YMax Then

YMax = Y(J)

I_at_YMax = I(J)

ElseIf Y(J) > 0 Then

MaxCount = J

End If

Next J

If SuperimposePlots = True Then

ResultsPicture.ScaleMode = 0

ResultsPicture.ScaleTop = OldYMax

ResultsPicture.ScaleHeight = -OldYMax

ResultsPicture.ScaleLeft = IatOldMinCount

ResultsPicture.ScaleWidth = IatOldMaxCount - IatOldMinCount

YMaxText.Text = Format\$(OldYMax, "0.00 e+00")

IMinText.Text = Format\$(OldIMin, "0.00 e+00")

IMaxText.Text = Format\$(OldIMax, "0.00 e+00")

Else

ResultsPicture.Cls

ResultsPicture.ScaleMode = 0

ResultsPicture.ScaleTop = YMax

ResultsPicture.ScaleHeight = -YMax

ResultsPicture.ScaleLeft = I(MinCount)

ResultsPicture.ScaleWidth = I(MaxCount) - I(MinCount)

YMaxText.Text = Format\$(YMax, "0.00 e+00")

IMinText.Text = Format\$(IMin, "0.00 e+00")

IMaxText.Text = Format\$(IMax, "0.00 e+00")

```

    OldYMax = YMax
    OldIMax = IMax
    OldIMin = IMin
    OldMaxCount = MaxCount
    IatOldMinCount = I(MinCount)
    IatOldMaxCount = I(MaxCount)
End If

For J = 1 To Samples

    ResultsPicture.Line (I(J - 1), Y(J - 1))-(I(J), Y(J))

Next J

I_at_YMaxDisplay.Text = Format$(I_at_YMax, "0.00 e+00")
DeltaTDisplay.Text = Format$(DeltaT, "0.00")

End Sub

Sub ExitName_Click ()
    Graph.Hide
End Sub

Sub PrintName_Click ()
    Graph.PrintForm
End Sub

Sub PerformanceName_Click ()
    PerformancePlot = True
    PerformanceName.Checked = True
    PumpName.Checked = False
End Sub

Sub PumpName_Click ()
    PerformancePlot = False
    PerformanceName.Checked = False
    PumpName.Checked = True
End Sub

Sub Form_Load ()
    PerformancePlot = True
    Samples = 200
End Sub

```

```
Sub DrawName_Click ()  
    Screen.MousePointer = 11  
    GraphIt  
    SuperName.Enabled = True  
    Screen.MousePointer = 0  
End Sub
```

```
Sub a100Name_Click ()  
    a100Name.Checked = True  
    a200Name.Checked = False  
    a500Name.Checked = False  
    a1000Name.Checked = False  
    Samples = 100  
End Sub
```

```
Sub a200Name_Click ()  
    a100Name.Checked = False  
    a200Name.Checked = True  
    a500Name.Checked = False  
    a1000Name.Checked = False  
    Samples = 200
```

```
End Sub
```

```
Sub a500Name_Click ()  
    a100Name.Checked = False  
    a200Name.Checked = False  
    a500Name.Checked = True  
    a1000Name.Checked = False  
    Samples = 500
```

```
End Sub
```

```
Sub a1000Name_Click ()  
    a100Name.Checked = False  
    a200Name.Checked = False  
    a500Name.Checked = False  
    a1000Name.Checked = True  
    Samples = 1000
```

```
End Sub
```

```
Sub SuperName_Click ()  
    If SuperName.Checked = False Then
```

```

        SuperimposePlots = True
        SuperName.Checked = True
    Else
        SuperimposePlots = False
        SuperName.Checked = False
    End If
End Sub

```

History.txt

```

// Cascaded Stages Window menu bar
Sub Form_Load ()
    History.Left = 0
    History.Top = Main.Height
End Sub

Sub PrintName_Click ()
    History.PrintForm
End Sub

```

Intro.txt

```

// Introduction graphic, program startup
Sub Timer1_Timer ()
    Screen.MousePointer = 11
    Load Results
    Load Performance
    Load Material
    Load Definitions
    Load TempCalc
    Load Graph
    Load Cascade
    Main.Show
    Load Cooler
    Load Ending
    Load History
    Intro.Hide
    Timer1.Enabled = 0
    Screen.MousePointer = 0
End Sub

Sub Form_Load ()
    Screen.MousePointer = 11
End Sub

```

Pumping.txt

// Code for Maximum Heat Pumped Window

Sub Command1_Click ()

Screen.MousePointer = 11

Stage = 0

Pumping.Hide

History.Hide

Results.Hide

Screen.MousePointer = 0

End Sub

Sub Form_Load ()

PerformanceCmd.Width = 2415

Pumping.Left = Screen.Width - Pumping.Width

Pumping.Top = Main.Height + 450

LowTempUpdate = True

Cascading = False

End Sub

Sub Command2_Click ()

Screen.MousePointer = 11

Command2.Caption = "Wait"

History.WindowState = 0

LowTempUpdate = True

Cascading = True

Stage = Stage + 1

If Stage = 1 Then

TotalPower = 0

End If

Tc = Val(LowTempText.Text)

Th = Val(HighTempDisplay.Text)

DeltaT = Th - Tc

DeltaTMax = .5 * z * (Tc ^ 2)

Th_plus_Tc = Th + Tc

gamma = Sqr(1 + (z / 2) * (Th_plus_Tc))

Io = (alpha * Tc) / R


```

Vo = alpha * Th
P = Io * Vo
qT = Val(ReferLoadText.Text)
qP = alpha * Io * Tc
qJ = .5 * (Io ^ 2) * R
qF = K * DeltaT
qo = .5 * ((alpha ^ 2 * Tc ^ 2) / R) - qF
No_of_Couples = qT / qo
Total_Heat_rejected = qT + No_of_Couples * P
Coeff_of_Performance = qo / P
TotalPower = TotalPower + (P * No_of_Couples)
If Stage = 1 Then
    FirstStage_qo = qo * No_of_Couples
End If
OverallCOP = FirstStage_qo / TotalPower

```

```

TMaxDisplay.Text = Format$(DeltaTMax + Tc, "##,##0.0")
TMinDisplay.Text = Format$(Tc, "##,##0.0")
DeltaTDisplay.Text = Format$(DeltaT, "##,##0.0")

```

```

Results.CoeffText.Text = Format$(Coeff_of_Performance, "0.0000")
Results.NumberCouplesText.Text = Format$(No_of_Couples, "#,##0.00")
Results.TotalHeatRejectedText.Text = Format$(Total_Heat_rejected, "#,##0.0")
Results.AperCoupleText.Text = Format$(Io, "#,##0.00")
Results.VperCoupleText.Text = Format$(Vo, "#,##0.0000")
Results.WperCoupleText.Text = Format$(P, "#,##0.0000")
Results.qTText.Text = Format$(qT, "#,##0.0")
Results.qPText.Text = Format$(qP, "#,##0.0000")
Results.qJText.Text = Format$(qJ, "#,##0.0000")
Results.qFText.Text = Format$(qF, "#,##0.0000")
Results.qoText.Text = Format$(qo, "#,##0.0000")
Results.Caption = "Maximum Heat Pumping" + " Stage " + Str$(Stage)
Results.Show

```

```

Select Case Stage

```

```

    Case 1

```

```

        History.qLoadStageOneDisplay.Text = Format$(qT, "#,##0.0")
        History.qRejectedStageOneDisplay.Text = Format$(Total_Heat_rejected,
"#,##0.0")
        History.CouplesStageOneDisplay.Text = Format$(No_of_Couples,
"#,##0.00")
        History.WperCoupleStageOneDisplay.Text = Format$(P, "#,##0.0000")
        History.TcStageOneDisplay.Text = Format$(Tc, "0.0000")
        History.ThStageOneDisplay.Text = Format$(Th, "0.0000")

```

```

History.DeltaTStageOneDisplay.Text = Format$(DeltaT, "0.0000")
History.StageCOPStageOneDisplay.Text = Format$(Coeff_of_Performance,
"0.0000")
History.OverallCOPStageOneDisplay.Text = Format$(OverallCOP, "0.0000")
History.Width = 2250
Case 2
History.qLoadStageTwoDisplay.Text = Format$(qT, "#,##0.0")
History.qRejectedStageTwoDisplay.Text = Format$(Total_Heat_rejected,
"#,##0.0")
History.CouplesStageTwoDisplay.Text = Format$(No_of_Couples,
"#,##0.00")
History.WperCoupleStageTwoDisplay.Text = Format$(P, "#,##0.0000")
History.TcStageTwoDisplay.Text = Format$(Tc, "0.0000")
History.ThStageTwoDisplay.Text = Format$(Th, "0.0000")
History.DeltaTStageTwoDisplay.Text = Format$(DeltaT, "0.0000")
History.StageCOPStageTwoDisplay.Text = Format$(Coeff_of_Performance,
"0.0000")
History.OverallCOPStageTwoDisplay.Text = Format$(OverallCOP, "0.0000")
History.Width = 3360
Case 3
History.qLoadStageThreeDisplay.Text = Format$(qT, "#,##0.0")
History.qRejectedStageThreeDisplay.Text = Format$(Total_Heat_rejected,
"#,##0.0")
History.CouplesStageThreeDisplay.Text = Format$(No_of_Couples,
"#,##0.00")
History.WperCoupleStageThreeDisplay.Text = Format$(P, "#,##0.0000")
History.TcStageThreeDisplay.Text = Format$(Tc, "0.0000")
History.ThStageThreeDisplay.Text = Format$(Th, "0.0000")
History.DeltaTStageThreeDisplay.Text = Format$(DeltaT, "0.0000")
History.StageCOPStageThreeDisplay.Text = Format$(Coeff_of_Performance,
"0.0000")
History.OverallCOPStageThreeDisplay.Text = Format$(OverallCOP,
"0.0000")
History.Width = 4425
Case 4
History.qLoadStageFourDisplay.Text = Format$(qT, "#,##0.0")
History.qRejectedStageFourDisplay.Text = Format$(Total_Heat_rejected,
"#,##0.0")
History.CouplesStageFourDisplay.Text = Format$(No_of_Couples,
"#,##0.00")
History.WperCoupleStageFourDisplay.Text = Format$(P, "#,##0.0000")
History.TcStageFourDisplay.Text = Format$(Tc, "0.0000")
History.ThStageFourDisplay.Text = Format$(Th, "0.0000")
History.DeltaTStageFourDisplay.Text = Format$(DeltaT, "0.0000")

```

```

        History.StageCOPStageFourDisplay.Text = Format$(Coeff_of_Performance,
"0.0000")
        History.OverallCOPStageFourDisplay.Text = Format$(OverallCOP, "0.0000")
        History.Width = 5520
    Case 5
        History.qLoadStageFiveDisplay.Text = Format$(qT, "###0.0")
        History.qRejectedStageFiveDisplay.Text = Format$(Total_Heat_rejected,
"###0.0")
        History.CouplesStageFiveDisplay.Text = Format$(No_of_Couples,
"###0.00")
        History.WperCoupleStageFiveDisplay.Text = Format$(P, "###0.0000")
        History.TcStageFiveDisplay.Text = Format$(Tc, "0.0000")
        History.ThStageFiveDisplay.Text = Format$(Th, "0.0000")
        History.DeltaTStageFiveDisplay.Text = Format$(DeltaT, "0.0000")
        History.StageCOPStageFiveDisplay.Text = Format$(Coeff_of_Performance,
"0.0000")
        History.OverallCOPStageFiveDisplay.Text = Format$(OverallCOP, "0.0000")
        History.Width = 6585
    End Select

    History.Show 0

    ScrollUpdate = False
    LowTempText.Text = Format$(Th, "###0.0")
    ScrollUpdate = True

    ReferLoadText.Text = Format$(Total_Heat_rejected, "###0.00")

    LowTempUpdate = False
    Tempscroll.Value = (Tc + DeltaT) * 10
    LowTempUpdate = True

    Command2.Caption = "Cascade"
    Screen.MousePointer = 0

    Cascading = False
End Sub

Sub TempScroll_Change ()
    If ScrollUpdate = True Then
        Tc = Val(LowTempText.Text)
        Th = Tempscroll.Value / 10
        DeltaT = Th - Tc
        HighTempDisplay.Text = Format$(Th, "###00.0")
    End If
End Sub

```

```

    DeltaTDisplay.Text = Format$(DeltaT, "##,##0.0")
End If
End Sub

```

```

Sub LowTempText_Change ()
If LowTempUpdate = True Then
    Tc = Val(LowTempText.Text)
    DeltaTMax = .5 * z * (Tc ^ 2)
    TMax = DeltaTMax + Tc

    Tempscroll.Max = Tc * 10 + 1
    Tempscroll.Min = TMax * 10 - 1
    If Stage = 0 Then
        Tempscroll.Value = Tempscroll.Min
    End If

```

```

    Th = Tempscroll.Value / 10

```

```

    If Cascading = False Then
        DeltaT = Th - Tc
    End If

```

```

    TMinDisplay.Text = Format$(Tc, "##,##0.0")
    TMaxDisplay.Text = Format$(TMax, "##,##0.0")
    DeltaTDisplay.Text = Format$(DeltaT, "##,##0.0")
End If
End Sub

```

```

Sub PerformanceCmd_Click ()

```

```

    Screen.MousePointer = 11
    Main.ShowGraphName.Enabled = True
    PerformanceCmd.Caption = "Wait"
    Command2.Visible = -1
    History.Hide

```

```

    Stage = 0

```

```

    DeltaT = Th - Tc
    Th_plus_Tc = Th + Tc
    gamma = Sqr(1 + (z / 2) * (Th_plus_Tc))

```

```

    Io = (alpha * Tc) / R
    Vo = alpha * Th

```

```

P = Io * Vo
qT = Val(ReferLoadText.Text)
qP = alpha * Io * Tc
qJ = .5 * (Io ^ 2) * R
qF = K * DeltaT
qo = .5 * ((alpha ^ 2 * Tc ^ 2) / R) - qF
No_of_Couples = qT / qo
Total_Heat_rejected = qT + No_of_Couples * P
Coeff_of_Performance = qo / P

```

```

TMaxDisplay.Text = Format$(DeltaTMax + Tc, "##,##0.0")
TMinDisplay.Text = Format$(Tc, "##,##0.0")
DeltaTDisplay.Text = Format$(DeltaT, "##,##0.0")

```

```

Results.CoeffText.Text = Format$(Coeff_of_Performance, "0.0000")
Results.NumberCouplesText.Text = Format$(No_of_Couples, "#,##0.00")
Results.TotalHeatRejectedText.Text = Format$(Total_Heat_rejected, "#,##0.0")
Results.AperCoupleText.Text = Format$(Io, "#,##0.00")
Results.VperCoupleText.Text = Format$(Vo, "#,##0.0000")
Results.WperCoupleText.Text = Format$(P, "#,##0.00")
Results.qTText.Text = Format$(qT, "#,##0.0")
Results.qPText.Text = Format$(qP, "#,##0.0000")
Results.qJText.Text = Format$(qJ, "#,##0.0000")
Results.qFText.Text = Format$(qF, "#,##0.0000")
Results.qoText.Text = Format$(qo, "#,##0.0000")
Results.Caption = "Maximum Heat Pumping"
Results.Show
Results.PowerConsumptionDisplay.Text = Format$((P * No_of_Couples),
"#,###,##0.00")
PerformanceCmd.Width = 975
PerformanceCmd.Caption = "Restart"
Screen.MousePointer = 0
End Sub

```

Results.txt

```

// Code for Results Window
Sub Form_Click ()
    Results.Top = (Screen.Height - Results.Height)
    Results.Left = (Screen.Width - Results.Width) / 2
End Sub

```



```

Sub Form_Load ()
    Results.Top = (Screen.Height - Results.Height)
    Results.Left = (Screen.Width - Results.Width) / 2
End Sub

```

```

Sub PrintName_Click ()
    Results.PrintForm
End Sub

```

Tempcalc.txt

// Code for Temperature Conversion Calculator

```

Sub Command2_Click ()
    ConvertDisplayText.Text = ""
    ConvertDisplayText.SetFocus
End Sub

```

```

Sub ConvertCmd_Click ()
    Const True = -1
    Const False = 0
    Temperature = Val(ConvertDisplayText.Text)
    If FarenheitOutOption.Value = True Then
        If RankineInOption.Value = True Then
            Temperature = Temperature - 459.67
        ElseIf FarenheitInOption.Value = True Then
            Temperature = Temperature
        ElseIf CelsiusInOption.Value = True Then
            Temperature = (Temperature * 9 / 5) + 32
        ElseIf KelvinInOption.Value = True Then
            Temperature = ((Temperature - 273.15) * 9 / 5) + 32
        End If
    End If

```

```

ElseIf RankineOutOption.Value = True Then

    If RankineInOption.Value = True Then
        Temperature = Temperature
    ElseIf FarenheitInOption.Value = True Then
        Temperature = Temperature + 459.67
    ElseIf CelsiusInOption.Value = True Then
        Temperature = ((Temperature) * 9 / 5) + 32 + 459.67
    ElseIf KelvinInOption.Value = True Then
        Temperature = ((Temperature - 273.15) * 9 / 5) + 32 + 459.67
    End If

```



```
ElseIf CelsiusOutOption.Value = True Then
```

```
    If RankineInOption.Value = True Then
```

```
        Temperature = (Temperature - 32 - 459.67) * 5 / 9
```

```
    ElseIf FarenheitInOption.Value = True Then
```

```
        Temperature = (Temperature - 32) * 5 / 9
```

```
    ElseIf CelsiusInOption.Value = True Then
```

```
        Temperature = Temperature
```

```
    ElseIf KelvinInOption.Value = True Then
```

```
        Temperature = Temperature - 273.15
```

```
    End If
```

```
ElseIf KelvinOutOption.Value = True Then
```

```
    If RankineInOption.Value = True Then
```

```
        Temperature = (Temperature - 32 - 459.67) * 5 / 9 + 273.15
```

```
    ElseIf FarenheitInOption.Value = True Then
```

```
        Temperature = (Temperature - 32) * 5 / 9 + 273.15
```

```
    ElseIf CelsiusInOption.Value = True Then
```

```
        Temperature = Temperature + 273.15
```

```
    ElseIf KelvinInOption.Value = True Then
```

```
        Temperature = Temperature
```

```
    End If
```

```
End If
```

```
ConvertDisplayText.Text = Format$(Temperature, "##,##0.00")
```

```
ConvertDisplayText.SetFocus
```

```
End Sub
```

```
Sub Form_Click ()
```

```
    ConvertDisplayText.SetFocus
```

```
End Sub
```

```
Sub Form_Load ()
```

```
    TempCalc.Left = (Screen.Width - TempCalc.Width) / 2
```

```
    TempCalc.Top = (Screen.Height - TempCalc.Height) / 2
```

```
End Sub
```

```
Sub ExitName_Click ()
```

```
    TempCalc.Hide
```

```
End Sub
```

LIST OF REFERENCES

1. Kraus, A. D., *Cooling Electronic Equipment*, Prentice-Hall, Inc., 1965.
2. Ioffe, A. F., *Semiconductor Thermoelements and Thermoelectric Cooling*, Infosearch Ltd., 1957.
3. Cadoff, I. B., *Thermoelectric Materials and Devices*, Reinhold, 1960.
4. Egli, P. H., and others, *Thermoelectricity*, John Wiley and Sons Inc., 1960.
5. Tipler, P. A., *Physics for Scientists and Engineers*, 3rd ed., Worth Publishers, 1991.

BIBLIOGRAPHY

1. Burshteyn, A. I., *Semiconductor Thermoelectric Devices*, Temple Press, 1964.
2. Craig, J. C., *The Microsoft Visual Basic Workshop*, Microsoft Press, 1991.
3. Microsoft Corporation, *Microsoft Visual Basic Language Reference*, 1991.
4. Microsoft Corporation, *Microsoft Visual Basic Programmer's Guide*, 1991.
5. Microsoft Corporation, *Microsoft Windows User's Guide*, Version 3.0, 1990.
6. Petzold, C., *Programming Windows 3.1*, 3rd ed., Microsoft Press, 1992.

INITIAL DISTRIBUTION LIST

- | | |
|---|---|
| 1. Defense Technical Information Center | 2 |
| Cameron Station | |
| Alexandria, Virginia 22304-6145 | |
| 2. Library, Code 0142 | 2 |
| Naval Postgraduate School | |
| Monterey, California 93943-5002 | |

814-455

DUDLEY M. LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTERE. 1070-5101



GAYLORD S



DUDLEY KNOX LIBRARY



3 2768 00018866 8